Been-Chian Chien
Tzung-Pei Hong
Shyi-Ming Chen
Moonis Ali (Eds.)

LNAI 5579

Next-Generation Applied Intelligence

22nd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2009 Tainan, Taiwan, June 2009, Proceedings

Springer

Lecture Notes in Artificial Intelligence 5579

Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

Been-Chian Chien Tzung-Pei Hong
Shyi-Ming Chen Moonis Ali (Eds.)

Next-Generation Applied Intelligence

22nd International Conference
on Industrial, Engineering and Other Applications
of Applied Intelligent Systems, IEA/AIE 2009
Tainan, Taiwan, June 24-27, 2009
Proceedings

Springer

Volume Editors

Been-Chian Chien
National University of Tainan
Tainan 700, Taiwan
E-mail: bcchien@mail.nutn.edu.tw

Tzung-Pei Hong
National University of Kaohsiung
Kaohsiung 811, Taiwan
E-mail: tphong@nuk.edu.tw

Shyi-Ming Chen
National Taiwan University of Science and Technology
Taipei, Taiwan
E-mail: smchen@mail.ntust.edu.tw

Moonis Ali
Texas State University-San Marcos
San Marcos, TX 78666-4616, USA
E-mail: ma04@txstate.edu

Library of Congress Control Number: Applied for

CR Subject Classification (1998): I.2, H.2.8, I.4, I.5, H.5.2, F.1, F.2

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-642-02567-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-02567-9 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12695751 06/3180 5 4 3 2 1 0

Preface

The International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE), always sponsored by the International Society of Applied Intelligence (ISAI), emphasizes applications of applied intelligent systems to solve real-life problems in all areas. It is held every year and has become one of the biggest and most important academic activities concerning the theory and applications of intelligent systems in the world. The IEA/AIE 2009 conference was hosted by the National University of Tainan and National University of Kaohsiung in Taiwan. This was the first time that the IEA/AIE conference was held in Taiwan.

We received 286 papers from all parts of the world. Only 84 papers were selected for publication in this volume of LNAI proceedings. Each paper was reviewed by at least two anonymous referees to assure the high quality. We would like to express our sincere thanks to the Program Committee members and all the reviewers for their hard work, which helped us to select the highest quality papers for the conference. These papers highlight opportunities and challenges for the next generation of applied intelligence and reveal technological innovations in real applications.

The papers in the proceedings cover the following topics: autonomous agents, computer vision, data mining and knowledge discovery, decision support systems, evolutionary computation, fuzzy logic and uncertainty, intelligent system applications, intelligent wireless computing, knowledge management and natural language processing, machine learning, multi-agent systems, and neural networks, as well as three special invited topics including engineering knowledge and semantic systems, mass customization of manufacturing products using applied intelligent systems, and mining interesting knowledge.

In addition to paper presentations, the conference invited five internationally famous scholars to give splendid keynote speeches. They were Hisao Ishibuchi from Osaka Prefecture University, Japan, Xin Yao from University of Birmingham, UK, Ngoc Thanh Nguyen from Wroclaw University of Technology, Poland, Wen-Lian Hsu from Academia Sinica, Taiwan, and Pau-Choo Chung from National Cheng Kung University, Taiwan.

We would like to thank our main sponsors, ISAI, as well as our other sponsors: Association for the Advancement of Artificial Intelligence (AAAI), Association for Computing Machinery (ACM/SIGART), Canadian Society for the Computational Studies of Intelligence (CSCSI/SCEIO), European Neural Network Society (ENNS), International Neural Network Society (INNS), Japanese Society for Artificial Intelligence (JSAI), The Society for Korean Institute of Intelligent Systems (KIIS), Taiwanese Association for Artificial Intelligence (TAAI), and Texas State University-San Marcos.

We wish to thank the National Science Council of Taiwan, R.O.C., the Ministry of Education of Taiwan, R.O.C., the Ministry of Economic Affairs of Taiwan, R.O.C., the National University of Tainan, the National University of Kaohsiung, the Tainan City Government, and the other academic sponsors for their financial and logistical support.

We would also like to thank the Special Session Chairs, Chao-Lin Liu and Ying Liu, and the special session organizers for their assistance in the special session invitation. We would like to thank Springer for their help in publishing the proceedings. Moreover, grateful appreciation is expressed to all the Organizing Committee members and the local staff, especially Chih-Chin Lai, Hsin-Chang Yang, Rong-Ming Chen, Tsung-Yen Chuang, Jiann-Shu Lee, Ming-Yi Ju and Mang-Chun Wang for their help. Without their efforts, this conference would not have been such a substantial achievement.

Finally, we cordially thank the other participants, as well as all the authors, who made important contributions to the organization of this conference. The conference would not have been possible without their valuable support.

March 2009

Been-Chian Chien
Tzung-Pei Hong
Shyi-Ming Chen
Moonis Ali

Organization

IEA/AIE 2009 was organized by the International Society of Applied Intelligence, the National University of Tainan, and the National University of Kaohsiung in cooperation with AAAI, ACM/SIGART, CSCSI/SCEIO, ENNS, INNS, JSAI, KIIS, TAAI and Texas State University-San Marcos.

General Chair

Moonis Ali	Texas State University, San Marcos, USA

Program Chairs

Shyi-Ming Chen	National Taiwan University of Science and Technology, Taiwan
Been-Chian Chien	National University of Tainan, Taiwan
Tzung-Pei Hong	National University of Kaohsiung, Taiwan

Organizing Committee

Finance Chairs

Rong-Ming Chen	National University of Tainan, Taiwan
Tang-Kai Yin	National University of Kaohsiung, Taiwan

Publication Chairs

Chih-Chin Lai	National University of Kaohsiung, Taiwan
Wen-Yang Lin	National University of Kaohsiung, Taiwan
Cheng-Jian Lin	National Chin-Yi University of Technology, Taiwan

Local Arrangements Chairs

Tzung-Shi Chen	National University of Tainan, Taiwan
Jiann-Shu Lee	National University of Tainan, Taiwan
Shing-Tai Pan	National University of Kaohsiung, Taiwan

Web Chairs

Chang-Shing Lee	National University of Tainan, Taiwan
Hsin-Chang Yang	National University of Kaohsiung, Taiwan

Publicity Chairs

Tsung-Yen Chuang	National University of Tainan, Taiwan
Yu-Hui Tao	National University of Kaohsiung, Taiwan
Shyue-Liang Wang	National University of Kaohsiung, Taiwan

Secretariat

Ming-Yi Ju	National University of Tainan, Taiwan
Chih-Hung Wu	National University of Kaohsiung, Taiwan

Special Session Chairs

Chao-Lin Liu	National Chengchi University, Taiwan
Ying Liu	The Hong Kong Polytechnic University, Hong Kong

Special Session Organizers

1. Engineering Knowledge and Semantic Systems
   Jason J. Jung (Yeungnam University, Korea)
   Dariusz Król (Wroclaw University of Technology, Poland)
2. Mass Customization of Manufacturing Products Using Applied Intelligent Systems
   Vincent Lee (Monash University, Australia)
   Yen Cheung (Monash University, Australia)
   Helana Scheepers (Monash University, Australia)
3. Mining Interesting Knowledge
   Li-Shiang Tsay (North Carolina A&T State University, USA)
   Zbigniew W. Ras (University of North Carolina at Charlotte, USA)

Invited Speakers

Pau-Choo Chung	National Cheng Kung University, Taiwan
Wen-Lian Hsu	Academia Sinica, Taiwan
Hisao Ishibuchi	Osaka Prefecture University, Japan
Ngoc Thanh Nguyen	Wroclaw University of Technology, Poland
Xin Yao	The University of Birmingham, UK

Program Committee

Alahakoon, Damminda, Australia
Bae, Youngchul, Korea
Belli, Fevzi, Germany
Bhatt, Mehul, Germany
Bø, Ketil, Norway
Borzemski, Leszek, Poland
Bosse, Tibor, The Netherlands
Brézillon, Patrick, France
Burke, Edmund, UK
Chan, Chien-Chung, USA
Chan, C. W., Hong Kong
Chang, Bao-Rong, Taiwan
Lee, Shie-Jue, Taiwan
Lin, Cha-Hwa, Taiwan
Lin, Tsau Young, USA
Loia, Vincenzo, Italy
Lu, Wen-Hsiang, Taiwan
Madani, Kurosh, France
Matthews, Manton M., USA
Mizoguchi, Riichiro, Japan
Monostori, Laszlo, Hungary
Murphey, Yi Lu, USA
Nishida, Toyoki, Japan
Nguyen, Ngoc Thanh, Poland

Chang, Chein-I, USA
Chen, Rung-Ching, Taiwan
Chou, Jyh-Horng, Taiwan
Chung, Paul W. H., UK
Clifton, David A, UK
Cordón, Oscar, Spain
Dapoigny, Richard, France
Din, Der-Rong, Taiwan
Dreyfus, Gerard, France
Esposito, Floriana, Italy
Fatima, Shaheen, UK
Foulloy, Laurent, France
Fyfe, Colin, UK
Guesgen, Hans Werner, New Zealand
Ha, Sung Ho, Korea
Hagras, Hani, UK
Hendtlass, Tim, Australia
Hirota, Kaoru, Japan
Hoogendoorn, Mark, The Netherlands
Hsu, Chun-Nan, Taiwan
Hsu, Hui-Huang, Taiwan
Hsu, Shun-Chin, Taiwan
Huang, Hsiang-Cheh, Taiwan
Huang, Yo-Ping, Taiwan
Huget, Phillipe M., France
Hung, Chih-Cheng, USA
Hwang, Gwo-Jen, Taiwan
Ishibuchi, Hisao, Japan
Ito, Takayuki, Japan
Iwata, Shuichi, Japan
Jacquenet, Francois, France
Katarzyniak, Radosław, Poland
Kinoshita, Tetsuo, Japan
Kozae, Abdoulmonem, Egypt
Kumar, Amruth N., USA
Kumova, Bora I., Turkey
Kuo, Huang-Cheng, Taiwan
Lee, Huey-Ming, Taiwan
Okuno, Hiroshi G, Japan
Osawa, Yukio, Japan
Pan, Jeng-Shyang, Taiwan
Park, Gyei-Kark, Korea
Potter, Don, USA
Prade, Henri, France
Rajmohan, M., India
Ramaswamy, Srini, USA
Rau, Hsin, Taiwan
Rayward-Smith, Victor J., UK
Sadok, Djamel F. H., Brazil
Sànchez-Marrè, Miquel, Spain
Selim, Hasan, Turkey
Shpitalni, Moshe, Israel
Soo, Von-Wen, Taiwan
Sueda, Naomichi, Japan
Sun, Koun-Tem, Taiwan
Suzuki, Kazuhiko, Japan
Tamir, Dan, USA
Tan, Ah-Hwee, Singapore
Thulasiram, Ruppa K., Canada
Tseng, Lin-Yu, Taiwan
Tseng, Vincent Shin-Mu, Taiwan
Tsumoto, Shusaku, Japan
Vadera, Sunil, UK
Valtorta, Marco, USA
Vancza, Jozsef, Hungary
Viharos, Zsolt Janos, Hungary
Wang, Lipo, Singapore
Wong, Kainam Thomas, UK
Yang, Chunsheng, Canada
Yang, Yubin, China
Yang, Don-Lin, Taiwan
Zhang, Qingfu, UK
Kumova, Bora I., Turkey
Fatima, Shaheen, UK
Selim, Hasan, Turkey

Program Committee of Special Session 1

Bothorel, Cecile, France
Cao, Longbing, Australia
Freitas, Fred, Brazil
Godoy, Daniela, Argentina
Golbeck, Jennifer, USA
Jatowt, Adam, Japan
Kim, Hong-Gee, Korea
Lanzenberger, Monika, Austria
Lower, Michal, Poland
Mika, Peter, Spain
Nguyen, Ngoc Thanh, Poland
Stoermer, Heiko, Italy

Jo, Geun-Sik, Korea
Juszczyszyn, Krzysztof, Poland
Tabakow, Iwan, Poland
Zhdanova, Anna V., Austria

Program Committee of Special Session 2

Chau, Rowena, Australia
Dang, Xuan Hong, Australia
Fung, Lance, Australia
Ong, Kok Leong, Australia
Ng, Wee Keong, Singapore
Taniar, David, Australia

Program Committee of Special Session 3

An, Aijun, Canada
Appice, Annalisa, Italy
Cremilleux, Bruno, France
Cuzzocrea, Alfredo, Italy
Dardzinska, Agnieszka, Poland
Deogun, Jitender S., USA
Elomaa, Tapio, Finland
Favre, Cécile, France
Greco, Salvatore, Italy
Hamilton, Howard J., Canada
Im, Seunghyun, USA
Kwiatkowska, Mila, Canada
Qureshi, Taimur, France
Rauch, Jan, Czech
Rybinski, Henryk, Poland
Skowron, Andrzej, Poland
Slezak, Dominik, Canada
Tsumoto Shusaku, Japan
Tzacheva, Angelina, USA
Velcin, Julien, France
Yao, Yiyu, Canada
Zighed, Djamel A., France

Additional Reviewers

Acampora, Giovanni, Italy
Ali, Sy, USA
Al Shabi, Waled Haz, France
Amarger, Veronique, France
Ayesh, Aladdin, UK
Braiek, Ezedine Ben, Tunisia
Cabani, Adnane, France
Chan, Wing Chi, Hong Kong
Chang, Chia-Hui, Taiwan
Chang, Chuan-Yu, Taiwan
Chang, Feng-Cheng, Taiwan
Chang, Yue-Shan, Taiwan
Chebira, Abdennasser, France
Chen, Chin-Ling, Taiwan
Chen, James C., Taiwan
Chen, Jr-Shian, Taiwan
Chen, Lieu-Hen, Taiwan
Chen, S.-H., Taiwan
Chen, Ying-ping, Taiwan
Chen, Yueh-Hong, Taiwan
Cheng, Pu-Jen, Taiwan
Kao, Hung-Yu, Taiwan
King, Steve, UK
Król, Dariusz, Poland
Kuo, Yau-Hwang, Taiwan
Langevin, Scott, USA
Lee, Chu-hui, Taiwan
Lin, Chin-Feng, Taiwan
Lin, Da-Tung, Taiwan
Lin, Ming-Yen, Taiwan
Liu, Alan, Taiwan
Liu, T.-K., Taiwan
Liu, Xiangjie, China
Lu, Tzu-Chuen, Taiwan
Mauro, Nicola Di, Italy
Meng, Qinggang, UK
Ohki, Hidehiro, Japan
Ohmoto, Yoshimasa, Japan
Okada, Shogo, Japan
Orihara, Ryohei, Japan
Ouyang, Chen-Sen, Taiwan
Sabourin, Christophe, France

Chu, Hung-Chi, Taiwan
Dawson, Christian, UK
Do, Ellen Yi-Luen, USA
Doong, Shing H., Taiwan
Du, Yingzi, USA
Hattori, Hiromitsu, Japan
Hinde, Chris, UK
Ho, Wen-Hsien, Taiwan
Hsieh, Cheng-Hsiung, Taiwan
Hsieh, Fu-Shiung, Taiwan
Hsieh, Ming-Che, Taiwan
Hsieh, Shang-Lin, Taiwan
Hsu, Chung-Chian, Taiwan
Hsueh, Nien-lin, Taiwan
Huang, Jen-peng, Taiwan
Huang, Jingshan, USA
Huang, Po-Tsang, Taiwan
Huang, Tien Tsai, Taiwan
Jea, Kuen-Fang, Taiwan
Jeyapaul, R., India
Jou, Yung-Tsan, Taiwan
Juang, Chia-Feng, Taiwan
Sadok, Djamel F. H., Brazil
Sakurai, Shigeaki, Japan
Sandnes, Frode Eika, Norway
Senatore, Sabrina, Italy
Shiang, Wei-Jung, Taiwan
Shiau, Jiun-Yan, Taiwan
Shih, Jau-Ling, Taiwan
Song, Hua, China
Sundaram, Srini, UK
Tsai, Jinn-Tsong, Taiwan
Tsai, Tienwei, Taiwan
Wahde, Mattias, Sweden
Walker, Jessie, USA
Wang, Chu-Fu, Taiwan
Wang, Jingsong, USA
Wang, Jenq-Haur, Taiwan
Watson, Ian, New Zealand
Wu, Shih-Hung, Taiwan
Yang, Horng-Chang, Taiwan
Yeh, Ralph J.-F., Taiwan
Zhou, Jing, China

Table of Contents

Autonomous Agents

Computer Vision

Data Mining and Knowledge Discovery

Decision Support Systems

Evolutionary Computation

Fuzzy Logic and Uncertainty

Integration Systems for Real Life Applications

Intelligent Systems

Intelligent Wireless Computing

Knowledge Management and Natural Language Processing

Machine Learning

Multi-Agent Systems

Neural Networks

Special Session: Engineering Knowledge and Semantic Systems

Special Session: Mass Customization of Manufacturing Products using Applied Intelligent Systems

Special Session: Mining Interesting Knowledge (I)

Special Session: Mining Interesting Knowledge (II)

A General-Purpose Method for Decision-Making in Autonomous Robots

Mattias Wahde

Department of Applied Mechanics, Chalmers University of Technology,
412 96 Göteborg, Sweden
mattias.wahde@chalmers.se

Abstract. In this paper, it is argued that the standard taxonomy of behavior selection is incomplete. In order to overcome the limitations of standard behavior selection, a novel method for decision-making, the extended utility function (EUF) method, has been developed. Based on the concept of utility as a common currency for decision-making, the method handles decision-making involving both cognitive processes and (motor) behaviors, and is applicable as a general-purpose framework for decision-making in autonomous robots (as well as software agents). The EUF method is introduced and described, and it is then illustrated by means of an example. Preliminary tests indicate that the method performs well, allowing users rapidly to set up a decision-making system.

Keywords: Intelligent systems, autonomous agents.

1 Introduction

Traditionally, decision-making in behavior-based robotics has been referred to as *behavior selection* or *action selection* (see e.g. [1] for a recent review). The standard taxonomy of behavior selection systems [2] identifies two main approaches to the problem. In *arbitration methods*, normally a single behavior controls the robot at any given time, other behaviors being inactive. By contrast, in *command fusion methods*, the action taken by a robot represents a weighted average of the actions suggested by several different behaviors. However, this taxonomy is incomplete. First of all, it is strongly directed towards motor behaviors, i.e. behaviors that use one or several motors in order to move the robot or a part thereof. Indeed, the entire approach of behavior-based robotics (BBR) has been criticized for its inability to generate solutions to anything other than simple toy problems. In BBR, one commonly ties action directly to sensing; in other words, not much (cognitive) processing occurs. Second, even though cooperative methods (such as, for example, potential field navigation [3]) allow a robot to generate an action as a weighted average of the suggestions from several elementary behaviors, the taxonomy described above does not include methods that allow, for example, one or several cognitive processes (”thinking”, ranging from very simple low-level skills to much more complex processes) to execute simultaneously with a motor behavior (”acting”).

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 1–10, 2009.

The purpose of this paper is to introduce a new approach to robotic decision-making, referred to as the *extended utility function* (EUF) *method*. The EUF method aims to overcome many of the limitations inherent in the methods defined in the framework of the taxonomy described above and to allow users of the method to define robots displaying sufficiently advanced overall behavior to be useful in industrial and other applications.

2 The EUF Method

The EUF method has been developed as an extension of the utility function (UF) method [4], which, being an arbitration method in the taxonomy described above, has mainly been used for handling behavior selection among motor behaviors. The field of behavior-based robotics includes a bewildering array of terms that are often used slightly differently by different authors. Thus, before describing the actual method, an attempt will be made to clarify the various terms used in connection with the EUF method.

2.1 Nomenclature

The term *robotic brain* will be used to describe the program (running in the processor(s) of the robot) responsible for making decisions and taking actions. The term *control system* is specifically *not* used, as it signifies the more limited systems defined in the field of classical control theory. This is not to say that classical control is irrelevant: On the contrary, in (motor) behaviors (see below), the basic actions taken are often formulated in the form of classical control systems, such as e.g. PID controllers. However, the decision-making of the robot (i.e. selecting behaviors for activation or de-activation) is clearly another matter, hence the use of the term *robotic brain*. It should also be noted that this term does *not* imply that neural networks are used by default. Certain behaviors *may* of course make use of neural networks, but the term robotic brain, as used in the EUF method, does not in itself indicate that such structures are employed. Thus, one may say that the term is more oriented towards the overall functionality of a brain rather than its detailed structure.

In the EUF method, the robotic brain consists of a decision-making system (described below) and a repertoire (i.e. a set) of *brain processes*. Three kinds of (brain) processes are defined in the EUF method: (i) *cognitive processes* that do not involve any motor action, (ii) *locomotive behaviors* that displace the entire robot, using its motors, and (iii) *movement behaviors* that carry out movements not related to locomotion, such as moving the arms or the head (if available on the robot), or any combination thereof. Locomotive and movement behaviors both constitute motor behaviors. Thus, what one normally thinks of as a behavior (in BBR) is a special case of a brain process in the EUF method. Typically, a robot defined in the EUF framework is equipped with several cognitive processes and several motor behaviors. As for the motor behaviors, most wheeled and legged robots are equipped with at least *some* locomotive behaviors. On the other hand, movement behaviors are commonly used in, say, humanoid robots, whereas wheeled robots (of the kind considered in the example below) without arms or other manipulators normally do not use movement behaviors. A few specific examples of brain processes are given in connection with the example presented in Sect. 4 below.

In addition to the repertoire of brain processes, a *decision-making system* is defined, which is responsible for activation and de-activation of the various brain processes, in response to the different situations encountered by the robot.

2.2 Brain Processes

First and foremost, the EUF method deals with decision-making, i.e. selecting brain processes for activation, rather than the problem of actually generating the brain processes in the first place. However, in order to apply the EUF method, one must first generate a set of brain processes that will be used in the robot. Thus, a set of basic brain processes, briefly described in Sect 4.1 below, have been defined for the purpose of testing the method. For the remainder of this section, it will be assumed that such a set of brain processes is available.

2.3 Description of the Method

In the EUF method, the decision-making is based on the concept of *utility*, which can be seen as a common currency used for weighing different brain processes against each other in any given situation. The concept of utility, which was originally formalized by von Neumann and Morgenstern [5], has been applied in fields as diverse as economics, ethology and robotics [6]. However, before describing the use of utility in the decision-making process, it is necessary to introduce the concept of *state variables*.

State Variables. A robot normally obtains information via its sensors or other input devices such as keyboards or touch screens. Some sensors, such as infrared (IR) sensors, provide a scalar reading that (in the case of IR sensors) can be used, for example, in proximity detection. Other sensors provide vector-valued readings (e.g. laser range finders) or matrix-valued readings (e.g. digital cameras). In principle, all those readings could be used for determining the state of the robot. However, the data flow would be massive. Taking a cue from biology, one may note the ability of the (human) brain to filter out irrelevant information, in such a way that a person will only be consciously aware of information that is likely to be relevant for assessing the situation at hand. Thus, in the EUF method, a *sensory preprocessing system* (SPS) is introduced, which maps the raw data obtained through all the sensors (or a subset thereof) to a manageable number of *state variables*.

A variety of sensory preprocessing methods can be envisioned, and the EUF method does not introduce any restrictions on the type of mappings used in the SPS. As a simple example, a state variable z_1 that measures proximity to obstacles may be defined as the average value of the distances measured over an angular range (say $[-0.25, 0.25]$ radians, relative to the direction of heading of the robot), using a laser range finder (LRF), assuming that the robot is equipped with such a sensor. An SPS normally contains many mappings, each of which produces a scalar output z_k. The state variables are then collected in a vector (denoted $\mathbf{z}$) containing all outputs from the SPS. This vector is used as input to the utility functions, which will be described next.

Utility functions. In the EUF method, each brain process is equipped with a *utility function* that determines the relative merit of the brain process in the current situation. The utility u_i of brain process i is determined as

$$\tau_i \dot{u}_i + u_i = \sigma_i \left(\sum_{k=1}^{m} a_{ik} z_k + b_i + \Gamma_i \right), \quad i = 1, \ldots, n, \tag{1}$$

where n is the number of brain processes, τ_i is a time constant determining the reaction time of the robot (typically set to around 0.1 s), m is the number of state variables[1], a_{ik} and b_i are tunable parameters (the procedure of setting parameters is exemplified in Sect. 4.2 below), and $\sigma_i(x)$ is taken as $\tanh(c_i x)$, where c_i is a positive constant. Thus, the squashing functions σ_i serve to keep utility values in the range $[-1, 1]$ provided, of course, that the values are initialized in this range.

The parameter Γ_i, which is normally equal to zero, allows direct activation or de-activation of a brain process. Ideally, the state variables z_k $(k = 1, \ldots, m)$ should provide the robot with all the information needed to make an informed decision regarding which brain processes to keep active in any situation encountered. In practice, of course, it is very difficult to summarize, in a set of scalar state variables, all the many situations that may occur. As an alternative, a brain process j may set the parameter Γ_i of some brain process i either to a large positive value (in order to raise the utility u_i, so as to activate process i) or a large negative value (to achieve the opposite effect, i.e. de-activation of process i). However, once the intended result has been achieved, Γ_i should return to its default value of zero. This is achieved by letting Γ_i vary (at all times) as

$$\tau_i^{\Gamma} \dot{\Gamma}_i = -\Gamma_i \tag{2}$$

where τ_i^{Γ} is a time constant[2], determining the decay rate of Γ_i. Thus, in the normal situation Γ_i is equal to zero, whereas if some brain process abruptly sets Γ_i to a value different from zero, it subsequently falls off exponentially.

[1] In practice, Eq. (1) is discretized and is integrated with a time step (typically around 0.01 s) much smaller than the smallest time constant. Note that the latest available state variables are used as inputs to the utility functions. Some state variables may change very frequently, whereas others (e.g. those based on LRF readings) are updated more seldom (typically with a frequency of 10 Hz, in the case of an LRF).

[2] The superscript (which is not an exponent!) is introduced in order to distinguish this time constant from τ_i defined in Eq. (1).

```
repeat
 t ← t + dt
 Update sensor readings
 Determine state variables, using the SPS
 Obtain non-zero $\Gamma_i$ parameters (if any) from the (active) brain processes
 Update the utility functions and the $\Gamma_i$ parameters
 Activate and de-activate brain processes based on the utility values
 Execute active processes (for a time interval of length dt)
until (Terminated)
```

Fig. 1. Pseudo-code summarizing the activities taking place in a robotic brain defined in the framework of the EUF method. SPS = sensory preprocessing system. See also Eqs. (1) and (2). The integration time step dt is set to a value smaller than the smallest time constant in the system.

The differential form of Eq. (1), has the important effect of filtering out the inevitable noise present in the sensor readings and therefore transmitted to the state variables.

Decision-Making. The generation of a specific decision-making system in the EUF method consists of setting the parameters of the utility functions, as well as specifying both the mappings constituting the SPS and the use (if any) of the Γ_i parameters, so as to achieve the desired overall result. In many cases, this is in fact easier than it sounds. However, in complex cases, it may be difficult to provide appropriate parameter values by hand, as noted also in connection with the original UF method [4]. One may then resort to optimization of the parameters, using stochastic optimization algorithms such as, for example, genetic algorithms or particle swarm optimization.

In any case, assuming that one has been able to find appropriate values for the parameters (either by hand or using some optimization method), the forms of the state variables and the utility functions are thus determined. At this stage, decision-making is quite simple in the EUF method, and works as follows: Any cognitive process with utility larger than zero is active. Thus, it is possible for several cognitive processes to be active at the same time. By contrast, exactly *one* locomotive behavior is active at any given time, motivated by the fact that the robot cannot move in two different directions at the same time. In the EUF method, the locomotive behavior with highest utility is active, and all other locomotive behaviors are inactive. Of course, this does not imply that the robot must constantly be on the move: A locomotive behavior may also allow the robot to stand still. The activation of movement behaviors (if available) works in the same way: Exactly one such behavior is active at any given time, namely the one with highest current utility among the movement behaviors. Note, however, that movement behaviors may, of course, combine the use of several (non-locomotive) motors. For example, in a robot consisting of a wheeled base and a humanoid upper body, a movement behavior may be responsible for turning the head (and, possibly, the eyes) as well as moving the arms, to carry out a hand-eye coordination task. At the same time, the active locomotive behavior may move the entire robot forward. It should also be noted that the utility functions of *all* brain processes are continuously updated, meaning that an inactive process can be activated, should its utility value become sufficiently high. The activities occurring in the brain of the robot are summarized in pseudo-code in Fig. 1.

3 Implementation

Even though the EUF method is defined irrespective of any particular implementation, in order to actually *use* the method one must, of course, provide an implementation, normally in the form of a computer program. As in any problem involving autonomous robots, one may implement the method either in a computer simulation or in the processor(s) of a real robot. Ultimately, it is the ability of the method to provide a decision-making system for *real* robots that is of importance. However, simulations are useful in the early stages of the development of an autonomous robot since, for example, the construction of the actual robot is normally a costly and time-consuming process. Using simulations, one may reduce the risk of making serious mistakes in the construction process such as, for example, fitting the robot with inappropriate sensors for the task at hand. Also, if the development of the robot (body or brain, or both) involves some form of optimization, the time required for carrying out such a procedure in a real robot is often prohibitively long. On the other hand one should be careful to remember that the results obtained in a simulator at best represent a very rough approximation of the performance of the corresponding real robot, and only if the simulator manages to capture (most of) the relevant aspects of the real world such as, for instance, noise in sensors and actuators.

Note that using simulations for evaluating the EUF method is well motivated, since the complexity of the actual decision-making process is no different in a simulation (with appropriate noise levels in sensors, actuators etc.) than in the real world. In addition, the EUF method may, in fact, be used for decision-making in software agents (in computer games, information systems etc.), further motivating the use of simulations.

The EUF method has been implemented in a simulation program. Written in Delphi (object-oriented Pascal), the simulator allows a user to set up a simulation involving a differentially steered robot equipped with a repertoire of brain processes as well as a decision-making system following the EUF method (with an SPS, utility functions etc., as described above). The simulator introduces appropriate noise levels in both actuators and sensors. The detailed description of the robotic brain (i.e. the brain process repertoire, the parameter values of the decision-making system, the SPS mappings etc.) must be provided, by the user, in the form of a text file. Similarly, the parameters of the robot (physical characteristics, motors, sensors etc.) must also be provided, in the same way. A third text file, describing the arena, should also be generated. As these files tend to become rather complex, a set of template files has been generated, so that the user normally only needs to make minor modifications to those files. Once the setup files have been generated, the simulation can be executed. The program allows 3D visualization using OpenGL, so that the user easily can follow

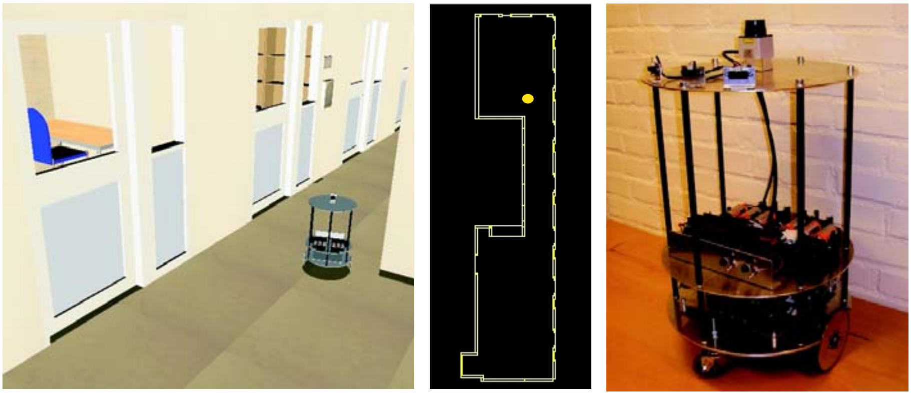


Fig. 2. Left panel: A screenshot from a simulation, showing a wheeled robot in a typical office environment. Middle panel: A schematic view (from above) of the office environment in which the simulations were carried out. Right panel: The physical counterpart (currently under construction) to the robot used in the simulations.

the performance of the robot, and also stop execution in order to modify, for example, the parameters determining the decision-making procedure.

An important aspect of the simulation is the possibility of emulating parallel processing, since the EUF method allows any number of cognitive processes to be active simultaneously, in parallel with one locomotive behavior (and, if available, one movement behavior). Thus, the simulator has been equipped with this feature, in such a way that all active brain processes are allowed to execute for dt s (the time step length) before an actual movement of the robot is carried out. Since the simulator typically runs much faster than real time, an illusion of parallel processing is easily obtained.

As for implementation in real robots, the simulation code is currently being translated to `C#`, which was deemed more suitable for this purpose. Here, the parallel execution of brain processes is handled using multi-threading, in which each brain processes runs as a separate thread (in Microsoft Windows).

4 Examples and Results

The EUF method is currently being tested and evaluated. A brief summary of such a test will now be given, in which a robot is tasked with navigating between two arbitrary points in a typical office arena. In practical applications, such a robot could be used, for example, in delivery tasks. The physical robot, shown in the right panel of Fig. 2, is currently being constructed. It is equipped with two (DC) motors, one for each drive wheel, and is supported underneath by two castor wheels. For sensing, the robot has been fitted (on its top plane) with a Hokuyo URG-04LX LRF and three infrared proximity detectors. It is also equipped with wheel encoders, providing the raw data for odometry (described below). The robotic brain will be implemented on a small laptop (not shown) placed on the robot. Here, only the results obtained from simulations will be described. These simulations (involving a simulated version of the robot just described) have been carried out mostly to test the EUF method, but also as a part of an iterative procedure, involving both simulations and hardware construction, aiding the process of designing the physical robot. A screenshot from the simulator is shown in the left panel of Fig. 2, and a schematic view of the arena is given in the middle panel of the same figure.

4.1 Simulation Setup

The brain of the (simulated) robot consisted of a decision-making system in the EUF framework. The behavioral repertoire contained five brain processes, namely *Potential field navigation* (B1), *Side collision avoidance* (B2), *Frontal collision avoidance* (B3), *Odometry* (B4), and *Localization* (B5). B4 is a cognitive process, whereas the other four are locomotive processes. In B1, the robot follows a potential field [3], with (stationary) obstacles defining bumps in the field and the goal defining a gently sloping valley attracting the robot. Hence, just like B2 and B3, B1 also handles obstacle avoidance to some extent. However, potential field navigation is dependent on accurate positioning. Thus, B2 and B3 are needed in cases where the robot's positioning (through the odometry process described below) is inaccurate so that, for example, the robot may be headed for, say, a wall, even though the potential field (based on the inaccurate position) tells it that the path is clear. B2 and B3 can also be used for avoiding collisions with moving obstacles (e.g. people) that are not included in the potential field. B2 handles situations in which, for example, the robot moves along an extended obstacle (e.g. a wall) in a slightly incorrect direction, i.e. towards the obstacle rather than parallel to it. By contrast, B3 is used for avoiding head-on (or near head-on) collisions. B4 generates estimates of position, velocity and heading angle, based on the readings of wheel encoders on the motors. The localization process (B5) is used in order to counteract the inevitable drift in the odometry. When activated, this process matches the latest available wide-angle readings from the LRF to a map provided (along with the potential field) to the robot. This behavior is, in fact, quite complex, and has been tested thoroughly both in simulations and in physical robots [7], using the Hokuyo URG-04LX LRF.

Thus, in order for the robot to solve its navigation task successfully, it must handle several instances of decision-making. For example, while its task is to reach the current target position as fast as possible, it must do so without hitting any obstacles. Thus, the robot must occasionally activate B2 or B3 even though their activation delays the robot's arrival at the navigation target. Furthermore, from time to time, the robot must also activate B5 in order to recalibrate the odometry. However, this should not be done too often, since B5 (in its current form) requires the robot to reach a standstill, thus causing another delay.

4.2 Decision-Making System

The state variables needed for the utility functions (see Eq. (1)), were obtained from an SPS that, in this case, contained three mappings, each formed as an

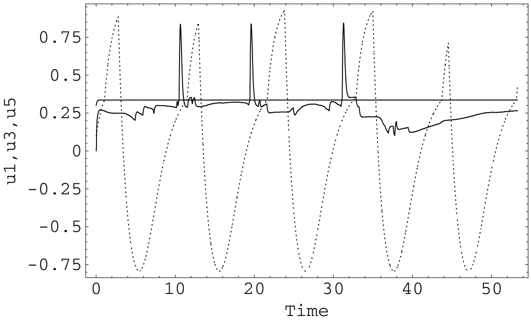


Fig. 3. The variation of utility values for brain processes B1 (solid, almost constant), B3 (solid, with spikes), and B5 (dotted), during a simulation

average over LRF readings. For example, the state variable z_1 was defined as the average of the distances detected by the LRF in the angular interval $[-0.25, 0.25]$. Thus, a low value of this variable would signal an imminent frontal collision. Using the three state variables, the five utility functions (one for each behavior) were set up. In the preliminary test described here, the parameters (i.e. a_{ik}, b_i etc.) were set by hand, which was possible in this rather simple example.

First of all, τ_1, τ_2, τ_3, and τ_4 were all set to 0.1 s, representing the reaction time of the robot. For B4, all a_{4k} were set to zero, and b_4 was set to a small positive value. The parameter Γ_4 was not used by any brain process. Hence, following Eq. (1), u_4 was always positive, meaning that the odometry process was constantly active. For the locomotive behaviors, where only the process with highest utility is active, it implies no restriction to set the utility of *one* such process to a constant value. Thus, the utility of B1 was set as that of B4. For the other three processes, some experimentation was needed and, in the end, B2 and B3 were, in fact, the only processes with non-zero a_{ik} parameters. For B5, τ_5 was set to 1.0 s. The bias (b_5) was set to a positive value, and the Γ_5 parameter was used to make sure that localization, when activated, was normally allowed to run to completion. Thus, upon activation, B5 would itself set Γ_5 to a large positive value, raising (within the next second or so) its utility. The fall-off of Γ_5 was regulated by setting the τ_5^{Γ} parameter to an appropriate value (2 s, in this case). In cases where B5 managed to complete its task rather quickly, instead of waiting for u_5 to drop, B5 would then essentially de-activate itself by setting Γ_5 to a large negative value. After only a few iterations involving different parameter settings, appropriate settings were found, and the robot was then capable of navigating reliably between arbitrary points in the arena.

An illustration of the robot's decision-making is shown in Fig. 3. The figure shows the variation in utility for B1, B3, and B5 over a simulation. For clarity, the utility functions for B2 and B4 have been omitted from the figure. In this particular run, B2 was never activated, whereas the cognitive process B4 (odometry) had positive utility throughout the run, and was therefore continuously active. The robot began by activating B4 and B1. B1 was then deactivated after around 1 s, when instead an initial localization (B5) was carried out. Next, the robot resumed its navigation until, after around 10 s, B1 was deactivated in favor of B3, in order to avoid a collision etc. Having covered the entire length of the arena, the robot reached its target after 53 s.

5 Discussion and Conclusion

The EUF method for decision-making in autonomous robots has been introduced and described. The preliminary tests carried out thus far are promising, and indicate that the EUF method allows a user to set up the decision-making structure quite rapidly, even though optimization is likely to be needed in complex cases. The use of differential equations for determining the utility functions effectively filters out the noise in the state variables. This represents an improvement over the earlier versions of the method (see e.g. [4]), where utility values were determined directly, i.e. without the derivative term present in Eq. (1). Current work involves implementation of the EUF method in physical robots, as well as the formulation of more complex test cases, involving even larger behavioral repertoires (including, for example, processes for human-robot interaction).

References

1. Bryson, J.J.: Mechanisms of action selection: Introduction to the special issue. Adaptive Behavior 15(1), 5–8 (2007)
2. Pirjanian, P.: Behavior-coordination mechanisms – state-of-the-art. Technical report IRIS-99-375, Institute for Robotics and Intelligent Systems, University of Southern California (1999)
3. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 500–505 (1985)
4. Wahde, M.: A method for behavioural organization for autonomous robots based on evolutionary optimization of utility functions. Journal of Systems and Control Engineering 217(4), 249–258 (2003)
5. von Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior, 3rd edn. Princeton University Press, Princeton (1953)
6. McFarland, D.: Animal Behaviour: Psychobiology, Ethology and Evolution, 3rd edn. Prentice-Hall, Englewood Cliffs (1998)
7. Pettersson, J., Hartono, P., Wahde, M.: A behavior module for odometry recalibration in autonomous robots. In: Proceedings of the Fourth International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE 2007), pp. 11–17 (2007)

A Virtual Human Agent Model with Behaviour Based on Feeling Exhaustion

Jan Treur

Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081, 1081 HV Amsterdam, The Netherlands
treur@cs.vu.nl
http://www.cs.vu.nl/~treur

Abstract. A computational agent model for monitoring and control of a virtual human agent's resources and exhaustion is presented. It models a physically grounded intelligent decision making process within the agent model for physical effort to be spent. Simulation results are discussed, and a formal analysis is presented on conditions under which the agent model functions properly, for example, such that it can be used to avoid running out of resources. Finally, the model is related to a model for monitoring or simulating a person's heart rate.

1 Introduction

To generate intelligent agent behaviour, it is more and more recognized that in addition to the brain, often the body plays a crucial role as well, and thus contributes to the intelligence. Some authors argue that also the design of artificial intelligent systems could gain benefit of such analyses by incorporating relevant physiological aspects in models developed; e.g., [2], [3], [5]. An example is the intelligence with which a person manages exhaustion (or fatigue). When the body would only give a signal when complete exhaustion occurs, no intelligent management would be possible. Fortunately, by gradually getting a feeling of becoming fatigued, more information is available before a total breakdown occurs. In this paper a computational model and analysis is made of the role that this physiological aspect (as described in more detail in the literature on physical exercise and sport) plays in monitoring and intelligent control of resources.

For certain types of artificial or virtual human agents, monitoring and control of the own resources may be of importance, for example, for human-like characters in a virtual reality context. When such a virtual human agent is equipped with the capability to get a feeling of becoming fatigued, it may show a more realistic intelligent behaviour. This behaviour can be based on a context-sensitive type of decision making to manage limitations of resources, incorporating the intelligence related to aspects of the body. Another application area is formed by ambient intelligence used in physical exercise and sport: devices that monitor human functioning, are able to analyse this functioning, and respond accordingly. When more sophisticated agent models are used, more advanced ambient intelligent agent applications can be developed.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 11–23, 2009.

In this paper literature on physical exercise and sport (e.g., [14]) is taken as a point of departure. One of the issues addressed is how the generated effort is controlled and what is the role of feeling fatigue in this process; e.g., [4], [6], [7], [8], [9], [10], [11], [12]. The interplay of mind and body in this process is considered a crucial factor. The classical perspective on fatigue is based on the assumption that fatigue occurs either when the muscles run out of resources, or they are in a sense poisoned by waste material produced; e.g., [8], [9]. Resources may involve oxygen, glycogen (which fuels the muscles), or ATP (adenosine triphosphate, the molecule that muscles use to store energy) that are lacking. Waste material concerns by-products of exercises, such as lactic acid. In this perspective the body reaches some states in which its functioning is disturbed so that only limited effort is possible, and this co-occurs with (or is expressed by) feeling fatigue. Noakes and his colleagues (e.g., [11], [12]) emphasize the notion of homeostasis: the property of a system (for example, a living organism), to regulate its internal environment in such a way that stable, more or less constant, conditions are maintained. In this view the mind keeps the body in physical conditions that are better prepared for expected or possible future efforts. A cognitive or neural decision making mechanism is assumed that incorporates information on the extent of exhaustion of the agent's body.

In Section 2 the computational agent model for monitoring and management of resources is presented and formalised. Section 3 discusses simulation results for the domain of cycling, which is used as a case study. In Section 4 a formal analysis is presented, which identifies the conditions and parameter values for which the model will function properly. Section 5 briefly analyses how the agent model can be related to a person's heart rate, which is sometimes used as monitor information in sports. Section 6 is a discussion.

2 The Computational Agent Model

This section describes the agent model for monitoring and control of resources. A central concept used is power that is generated. The basic idea behind the model is that it is easier to monitor the generated power at any time point, than the store of resources left. When based on monitor data on the generated power, the brain performs some form of accumulation or integration, then this can be used as a faithful indicator for the resources used. Within the literature on exercise and sports the notion of *critical power CP* plays an important role. This is the maximal level of power that can be generated and sustained over longer periods without becoming exhausted, assuming no prior exercising. It is an asymptote of the hyperbolic power-duration curve defined by $(GP - CP).t = W'$ that (as shown in various experiments) models the relationship between a constantly generated power GP (above the critical power CP) and the time t that this can be sustained; e.g., [6], [7], [8], [9]. Here W' is the total amount of work that can be spent above the critical power (the availble stored resources). Often it is (implicitly or explicitly) assumed that this critical power CP is a constant, that is not affected by prior exercising, and is a capacity to provide power based on aerobic processes. Power generated above this critical power is assumed to be based on anaerobic processes, that exploit an available fixed reservoir of stored basic resources *BR*, which is one of the parameters of the hyperbolic power-duration curve (often indicated by *W'*). Experiments show that after highly intensive prior exercising leading to exhaustion of the basic resources *BR*, for example, power at 90% of the critical power *CP* cannot be sustained; e.g., [4]. Therefore a main assumption made for the model developed below is that a *basic critical power BCP* (applicable when no prior exercising took place) can be distinguished from a *dynamic critical power DCP* (applicable when prior exercising has taken place). It is assumed that the dynamic critical power can vary between an upper bound (the basic critical power *BCP*) and a lower bound (*lowest critical power LCP*).

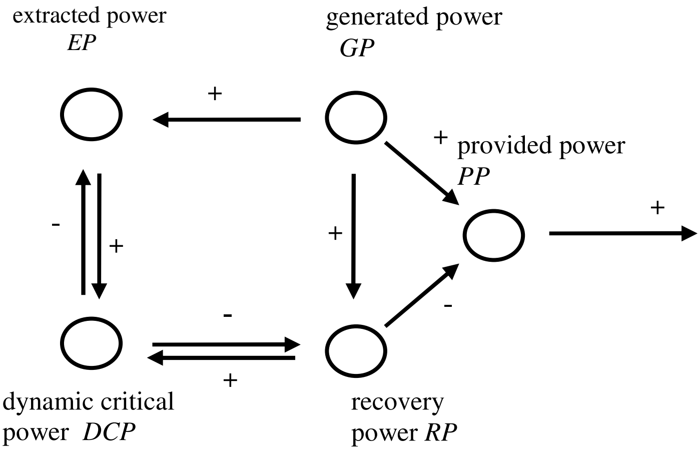


Fig. 1. Overview of the variables and their dependencies

Part of the dynamics of the model concerns how this dynamic critical power is affected by the effort spent above the critical power in the preceeding time interval (i.e., the extent to which the basic resources *BR* were already used), and how recovery can take place when the effort spent is below the critical power. Because the dynamic critical power has a direct relationship with the history of effort spent, it can be compared to the feeling of becoming fatigued, which indicates the extent of exhaustion, or the state of the (remaining) resources. In this way the dynamic critical power can be taken as a monitoring instrument to maintain homeostasis. Based on this indicator, decisions can be made on generated power. A possible decision model, for example, takes care that the dynamic critical power always is kept above a certain lower bound. When it is assumed that the dynamic critical power relates to fatigue (the lower the dynamic critical power, the more fatigue), this means that fatigue is kept limited. To obtain a formal model, the concepts used are formalised by numerical variables. Power is the amount of energy spent per time unit (expressed in Watt). Different types of power are distinguished; see Fig. 1 for the global effects they have on each other. Note that here the state *DCP* has a temporal relation to the other nodes, whereas the others have state relations; below more specific formulae are discussed.

generated power and dynamic critical power → extracted power

The *generated power* is the power level chosen by the person. If this is more than can be produced in a direct manner (by the aerobic system), part of this power is *extracted* from the (longer term) resources (the anaerobic system). When generated power is above the dynamic critical power, then the difference is extracted from the resources, indicated by:

$$EP(t) = Pos(GP(t) - DCP(t))$$

where the operator *Pos* for the positive part is defined by $Pos(x) = (x + |x|)/2$, or, alternatively: $Pos(x)= x$ if $x\geq 0$ and 0 else.

generated power and dynamic critical power → recovery power

When the generated power is lower than the dynamic critical power, a fraction of the generated power is contributed as recovery power. It is assumed that this recovery power is proportional to the difference between generated power and dynamic critical power, and also proportional to the difference between dynamic critical power and basic critical power. The proportion factor is *β*.

$$RP(t) = \beta\, GP(t)\, Pos(DCP(t) - GP(t)) \frac{BCP - DCP(t)}{BCP}$$

extracted power and recovery power → dynamic critical power

Extracted power decreases the dynamic critical power. Recovery power increases the dynamic critical power, bounded upward by the basic critical power *BCP*. The adjustment of the dynamic critical power after a time interval from t to $t+\Delta t$ is assumed proportional to the recovery power (factor γ_1), respectively the extracted power (factor γ_2).

$$DCP(t+\Delta t) = DCP(t) + (\gamma_1\, RP(t) - \gamma_2\, EP(t))\, \Delta t$$

The continuous model can be described by a differential equation:

$$\frac{dDCP(t)}{dt} = \gamma_1\, RP(t) - \gamma_2\, EP(t)$$

$$= \gamma_1\, \beta\, GP(t)\, Pos(DCP(t) - GP(t)) \frac{BCP - DCP(t)}{BCP} - \gamma_2\, Pos(GP(t) - DCP(t))$$

recovery power and generated power → provided power

The provided power is the difference between generated power and recovery power.

$$PP(t) = GP(t) - RP(t)$$

dynamic critical power → dynamic maximal power

The notion of maximal power models a limitation on the choice of of the generated power. The higher dynamic critical power, the higher the dynamic maximal power. Extracted power can only be positive as long as the dynamic critical power is above its minimum value *LCP*. The maximum power possible is assumed to be the dynamic critical power plus a constant *C*, as long as $DCP > LCP$ (no complete exhaustion), and equal to *LCP* when $DCP = LCP$ (complete exhaustion).

$$DMP(t) = C + DCP(t) \text{ when } DCP > LCP$$

$$DMP(t) = LCP(t) \qquad \text{when } DCP = LCP$$

$$BMP = C + BCP$$

Note that before reaching complete exhaustion, maximal power is substantially above the dynamic critical power, but upon reaching complete exhaustion, the maximal power drops to the level of the critical power, in accordance with the experiments reported, for example, in [4]. The agent model as described provides possibilities to make decisions based on the dynamic critical power as an indicator. When it is assumed that the dynamic critical power represents the feeling of fatigue, that feeling is in fact the indicator.

3 An Example Simulation

First a model for a cycling case study is described. In this case study the *provided power* is used to move a bike with a certain speed, depending on the resistance. Mechanical resistance can be taken into account in a *cycling efficiency factor*, for the process of generating *actual power* to move the bike. Further resistance is mainly based on air resistance, and if the road is ascending or descending on gravitation resistance. Given these resistances, *velocities* can be determined, and from them *distances*. For the sake of simplicity no gravitation resistance is considered; air resistance depends on a parameter called *air resistance coefficient*. To obtain a formal model, numerical variables are used: cycling efficiency factor *CEF*, air resistance coefficient *ARC,* actual cycling power *ACP*, and velocity *v*. The actual cycling power is the cycling efficiency factor times the provided power: $ACP(t) = CEF*PP(t)$. It is assumed that power exerted for movement is used to work against air resistance. For air resistance it is assumed that it is a force proportional to the square of velocity, with resistance coefficient *ARC*. Actual cycling power is the work performed per time unit, which is equal to this resistance force times the distance covered divided by the time (which is the velocity); therefore $ACP(t) = ARC*v(t)^3$ or:

$$v(t) = \sqrt[3]{\frac{ACP(t)}{ARC}}$$

Based on the model described in Section 2 and the cycling model described above, a number of simulation experiments have been performed, using existing numerical simulation tools. In Figure 2 results are shown of one of them, with time scale displayed in minutes; the step size Δt was taken one minute. The fixed parameter settings are: $\beta = 0.02$, $\gamma_1 = \gamma_2 = 0.4$, $\rho = 0.8$, $BCP = 400$, $LCP = 300$, and $C = 100$. The story goes as follows. First the cyclist generates power a bit above (*403*) the dynamic critical power (initially *400*), riding alone (air resistance coefficient *0.3*). The dynamic critical power slightly decreases. Then she joins a group of cyclists that passes by, and hence has less air resistance (coefficient *0.25*). The generated power is now lower than the dynamic critical power, while the speed is higher. Some recovery takes place. After a while in the group she is persuaded that it is her turn to take the front position. Now she has higher air resistance again, upon which she generates higher power (but lower speed); this brings the dynamic critical power down to near *350*. After some time she leaves the front position to somebody else, and while still being in the group she has less air resistance. However, as the dynamic critical power was decreased, the generated power now is still above the dynamic critical power; the dynamic critical power continues to decrease and approaches *300*: she cannot maintain the speed of this group after her effort at the front. She decides to leave the group and take some recovery time, riding on her own, with much lower speed, with more air resistance. The dynamic critical power increases; when another group passes by, she joins this

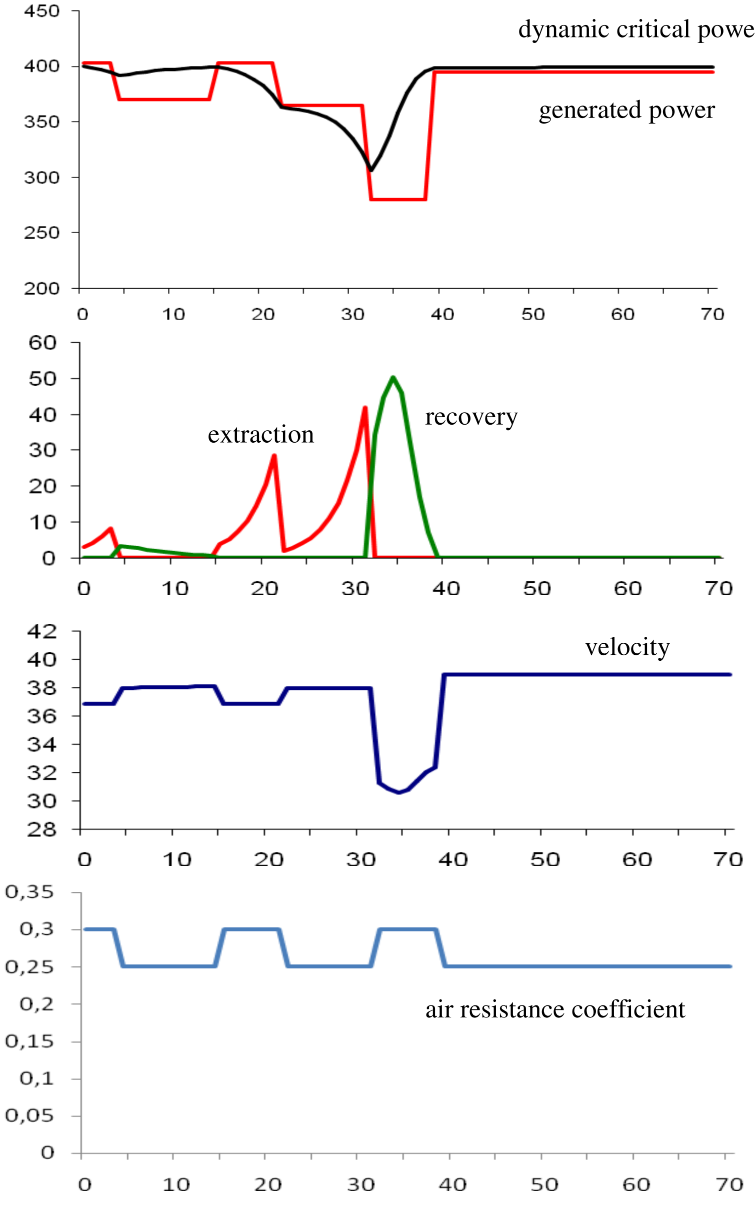


Fig. 2. Example simulation trace for the cycling case

group, and has lower air resistance again. Now the generated power stays slightly under the dynamic critical power, although this group has a higher speed. A state occurs with (almost) constant dynamic critical power.

Note that after 30 minutes a critical situation occurred. Taking the front position in the (first) group took so much of the resources that after leaving the front position, still the generated power to stay within the group was higher than the dynamic critical power. Therefore the cyclist could not recover and instead had to extract more and more from the resources, bringing the dynamic critical power further down. If no decision would have been made to decrease the generated power, within a very short time the decreasing trend of the dynamic critical power would have led to the lower bound *LCP* of the dynamic critical power, after which the maximal generated power would drop to the level of *LCP*.

The decisions of the cyclist have been modelled by the generated power levels over time, set by hand (see the lowest graph in Fig. 2). Note that in the second picture in Fig. 2, the areas under the extraction graph and the recovery graph are more or less the same, which means that the dynamic critical power after 40 minutes is almost equal to the one at the start, which also can be seen in the first graph.

4 Formal Analysis

A main question addressed in the formal analysis is whether the introduced agent model allows the person to monitor and control its resources in a proper manner. For proof sketches, see Appendix A.

Maintaining a steady state

Enduration sporters often try to maintain what they call a steady state: a state in which the main parameters are kept constant. In particular, it can be analysed how $DCP(t)$ can be kept constant. From the differential equation for $DCP(t)$ it follows:

$$\frac{dDCP(t)}{dt} = \gamma_1\, \beta\, GP(t)\, Pos(DCP(t) - GP(t))\, \frac{BCP - DCP(t)}{BCP} - \gamma_2\, Pos(GP(t) - DCP(t))$$

Since never both $Pos(x)$ and $Pos(-x)$ are nonzero, $\frac{dDCP(t)}{dt} = 0$ is equivalent to

$$GP(t)\, Pos(DCP(t) - GP(t))\, \frac{BCP - DCP(t)}{BCP} = 0 \quad \text{and} \quad Pos(GP(t) - DCP(t)) = 0$$

This is logically equivalent to

either $GP(t) = 0$ and $Pos(GP(t) - DCP(t)) = 0$
or $Pos(DCP(t) - GP(t)) = 0$ and $Pos(GP(t) - DCP(t)) = 0$
or $\frac{BCP - DCP(t)}{BCP} = 0$ and $Pos(GP(t) - DCP(t)) = 0$

which can be rewritten to

either $GP(t) = 0$ and $DCP(t) \geq 0$
or $GP(t) = DCP(t)$
or $DCP(t) = BCP$ and $GP(t) \leq DCP(t)$

This analysis is summarised in the following theorem expressing how a steady state for $DCP(t)$ can be characterised.

Theorem 1 (Characterising a Steady State)
For any time point *t*, when $GP(t) > 0$, the following are equivalent:

(i) $\frac{dDCP(t)}{dt} = 0$ (dynamic critical power equilibrium)
(ii) Either $GP(t) = DCP(t)$ or $GP(t) \leq DCP(t) = BCP$

How the indicator relates to the resources
A next issues to address in the analysis is the question how the dynamic critical power relates to the real resources. The level of resources $R(t)$ is assumed to be based on losses per time unit that are proportional to the extracted power (factor α_2) and gains proportional to the recovery power (factor α_1):

$$\frac{dR(t)}{dt} = \alpha_1 RP(t) - \alpha_2 EP(t)$$

In a different form this can be expressed by.

$$R(t_2) = R(t_1) + \alpha_1 \int_{t1}^{t2} RP(t)dt - \alpha_2 \int_{t1}^{t2} EP(t)dt$$

Recall that by BR the basic (additional) resources are denoted (in the literature often called W'). When at time point t_0 the resources are the basic resources, then the model can be described as:

$$R(t) = BR + \alpha_1 \int_{t0}^{t} RP(t)dt - \alpha_2 \int_{t0}^{t} EP(t)dt$$

To analyse the relationships between the indicators and the resource level, two special extreme cases are considered first: the case that the values are equal to the basic values, as in a situation at rest (addressed in Proposition 1), and the case that the resource level is *0*, as in a situation with completely exhausted resources (addressed in Proposition 2). For proper functioning in both special cases certain conditions on the parameters are identified in the two propositions. These conditions also turn out sufficient for proper functioning for the general case, as covered by Theorem 2.

In Proposition 1 it is investigated under which conditions the dynamic critical power is a proper indicator for the basic resources. It turns out that this is the case under a certain condition on the parameters; the same condition implies that the resource level is a linear function of the dynamic critical power. The proposition makes use of Lemma 1 which related the dynamic critical power power to the resource levels.

Lemma 1 (Critical Power vs Resource Level)
For all time points t_1 and t_2 it holds:

$$\gamma_1 (R(t_2) - R(t_1)) = \alpha_1 (DCP(t_2) - DCP(t_1)) + (\alpha_1\gamma_2 - \gamma_1\alpha_2) \int_{t1}^{t2} EP(t)dt$$

Proposition 1 (Indicating Basic Resource Levels)
The following are equivalent:

(i) The dynamic critical power is a proper indicator for basic resources:

$$\forall t \; [\, DCP(t) = BCP \Leftrightarrow R(t) = BR \,]$$

(ii) $\gamma_1/\alpha_1 = \gamma_2/\alpha_2$
(iii) For all time points t_1 and t_2 it holds:

$$\alpha_1 (DCP(t_2) - DCP(t_1)) = \gamma_1 (R(t_2) - R(t_1))$$

(iv) The expression $\gamma_1\ R(t) - \alpha_1\ DCP(t)$ is invariant over time: for all time points t_1 and t_2 it holds:

$$\gamma_1\ R(t_2) - \alpha_1\ DCP(t_2) = \gamma_1 R(t_1) - \alpha_1\ DCP(t_1)$$

(v) The dynamic critical power is a linear function of the resource level: for a given point t_0, for all time points t it holds:

$$\alpha_1\ DCP(t) = \gamma_1\ R(t) + \alpha_1\ DCP(t_0)\ \ -\ \ \gamma_1 R(t_0)$$

In Proposition 2 it is investigated under which conditions the dynamic critical power is a proper indicator for running out of resources. This is the case under a certain further condition on the parameters; this condition implies that the resource level is proportional to the dynamic critical power, as expressed in Theorem 2. Below it is assumed that at the initial time point t_0 the resources $BR(t_0)$ are the basic resources BR and the dynamic critical power $DCP(t_0)$ is the basic critical power BCP.

Proposition 2 (Indicating Running Out of Resources)
Suppose $\gamma_1/\alpha_1 = \gamma_2/\alpha_2$ and let $\eta = BCP/BR$. Then the following are equivalent:

(i) The dynamic critical power is a proper indicator for running out of resources:

$$\forall t\ \ [\ DCP(t) = LCP \ \ \Leftrightarrow\ \ R(t) = 0\ ]$$

(ii) At any point of time the dynamic critical power as an indicator faithfully (proportionally) reflects the resources left:
$DCP(t)\ /\ R(t)$ is constant over time

(iii) $\gamma_1/\alpha_1 = \gamma_2/\alpha_2 = \eta$

Theorem 2 (Proper Indicator for Resources)
In case $\gamma_1/\alpha_1 = \gamma_2/\alpha_2 = \eta$, with $\eta = BCP/BR$, the following hold:

a) The dynamic critical power is a proper general indicator for the resource level. More specifically, $DCP(t)$ is proportional to $R(t)$ over time with η as factor:

$$\forall t\ \ \ DCP(t) = \eta\ R(t)$$

b) Suppose for some lower bound $L{>}LCP$ the basic power power $DCP(t)$ is always kept above L i.e., $\forall t\ DCP(t) \geq L$, then the person will never run out of resources.

c) When the person has nonzero resources and $DCP(t)$ is kept constant, then the person will never run out of resources.

In this theorem a) and b) immediately follow from Proposition 2. Concerning c), when the person keeps $DCP(t)$ at a constant value above LCP, then b) applies. Therefore the person can only run out of resources when $DCP(t) = LCP$, but as $DCP(t)$ is kept constant, then by Proposition 2 there already were no resources, which is not the case.

Corollary
In case $\gamma_1/\alpha_1 = \gamma_2/\alpha_2 = \eta$, with $\eta = BCP/BR$, the resource level $R(t)$ over time satisfies the following differential equation

$$\frac{dR(t)}{dt} = \alpha_1\ \beta\ GP(t)\ Pos(\eta\ R(t) - GP(t))\ \frac{BR - R(t)}{BR}\ \ - \alpha_2\ Pos(GP(t) - \eta\ R(t))$$

This corollary follows from the differential equation

$$\frac{dR(t)}{dt} = \alpha_1\, RP(t) - \alpha_2\, EP(t)$$

for $R(t)$ by expressing $EP(t)$ and $RP(t)$ in $GP(t)$ and $DCP(t)$ and then, according to Theorem 2, replacing $DCP(t)$ by $\eta\, R(t)$. An alternative derivation is by taking the differential equation for $DCP(t)$ and replacing $DCP(t)$ by $\eta\, R(t)$.

Notice that the parameters α_1, α_2 are physiological parameters related to mechanisms for extraction from and recovery of resources. In contrast the parameters γ_1, γ_2 are neural or cognitive parameters, assumed to be set by the brain. To obtain a faithful indicator system, the relationships between the physiological and neural parameters as expressed in Theorem 2 are needed. It is assumed that within the brain the neural parameters stand in these relationships to the physiological parameters.

5 Relation to Heart Rate

In exercising and sport practice sometimes not (only) the feeling of fatigue, but (also) the heart rate is used to monitor the extent of exhaustion; here often some estimations are used, assuming a linear relation of heart rate to generated power above the critical power point. For example, if it is assumed that the heart rate for generated power at the level of the critical power is a constant *BHR*, and the maximal heart rate is a constant *MHR*, then for cases where $GP(t)$ is at least $DCP(t)$ the heart rate can be estimated as:

$$\begin{aligned} HR(t) &= BHR + (MHR–BHR).(GP(t)–DCP(t))/(DMP(t)–DCP(t)) \\ &= BHR + (MHR–BHR)/(GP(t)–DCP(t))/C = BHR + \gamma(GP(t)–DCP(t)) \end{aligned}$$

with $\gamma = (MHR–BHR)/C$. From this relation it follows that

$$\frac{dHR(t)}{dt} = \gamma\,\frac{dGP(t)}{dt} - \gamma\,\frac{dDCP(t)}{dt}$$

Therefore in case of generated power in equilibrium (for example, constant), the following can be derived.

Theorem 3 (Heart Rate as Indicator)

a) If $GP(t) \geq DCP(t)$ and it holds $\frac{dGP(t)}{dt} = 0$ (generated power equilibrium) then the following are equivalent:

(i) $\frac{dHR(t)}{dt} = 0$ (heart rate equilibrium)

(ii) $\frac{dDCP(t)}{dt} = 0$ (critical power equilibrium)

b) When both the generated power and heart rate are constant, then no running out of resources will occur.

6 Discussion

In this paper a virtual human agent model was introduced that addresses the notion of critical power which plays a central role in scientific and practical literature on exercising and sport. It performs integration of monitoring data on generated power over time and uses this to determine in a dynamic manner the critical power as an indicator for the amount of stored resources left. The model realises homeostasis under the assumption that the human uses this indicator to make the proper decisions about its generated power. Decision criteria for generated power are, for example, keeping this indicator constant (achieving a steady state), or keeping the indicator between certain bounds, thus avoiding running out of resources. It has also been discussed how a person's heart rate can be related to the dynamic critical power and used as an indicator. The model has been used to perform a number of simulations, one of which was presented above. Moreover, a formal analysis has been undertaken that shows under which conditions the critical power indicator indeed correlates to what it is expected to indicate, and that some strategies often used are guaranteed to work out well. A first test of the model against empirical data has shown some preliminary results which were positive.

The agent model may be useful in a number of application areas for intelligent agents. In the first place it may by useful as a basis for virtual characters with a realistic appearance (which, for example, may show a heart rate depending on efforts made). Furthermore, the agent model may be applied as part of an intelligent ambient agent device interacting with humans in physical exercise or sport, or in other demanding circumstances. Supporting devices mostly they concentrate on the sensoring and do not possess much intelligence to perform analysis of sensor data. Incorporating a dynamical model may provide a basis for more intelligent ambient agents. As an example of this development, some of the ideas reported in the current paper already were the inspiration for a part of the functional state model described in [1]. Finally, the model may be used for social simulations, to investigate how persons can cooperate in order to manage their resources more economically. For example, in cycling sports, based on riding behind each other sophisticated cooperation strategies between competing teams are followed, including negotiation processes.

References

1. Bosse, T., Both, F., van Lambalgen, R., Treur, J.: An Agent Model for a Human's Functional State and Performance. In: Jain, L., Gini, M., Faltings, B.B., Terano, T., Zhang, C., Cercone, N., Cao, L. (eds.) Proc. of the 8th IEEE/WIC/ACM Intern.Conf. on Intelligent Agent Technology, IAT 2008, pp. 302–307. IEEE Computer Society Press, Los Alamitos (2008)
2. Cannon, W.B.: The Wisdom of the Body. W.W. Norton and Co., New York (1932)
3. Clark, A.: Being There: Putting Brain Body and World Together Again. MIT Press, Cambridge (1997)
4. Coats, E.M., Rossiter, H.B., Day, J.R., Miura, A., Fukuba, Y., Whipp, B.J.: Intensity-dependent tolerance to exercise after attaining $V^{\cdot}_{O2\ max}$ in humans. J. of Applied Physiology 95, 483–490 (2003)
5. Damasio, A.: The Feeling of What Happens: Body, Emotion and the Making of Consciousness. MIT Press, Cambridge (1999)
6. Fukuba, Y., Whipp, B.J.: A metabolic limit on the ability to make up for lost time in endurance events. J. of Applied Physiology 87, 853–861 (1999)
7. Fukuba, Y., Miura, A., Endo, M., Kan, A., Yanagawa, K., Whipp, B.J.: The Curvature Constant Parameter of the Power-Duration Curve for Varied-Power Exercise. Medicine and Science in Sports and Exercise 35, 1413–1418 (2003)

8. Hill, A.V., Long, C.N.V., Lupton, H.: Muscular exercise, lactic acid, and the supply and utilisation of oxygen. Proc. Royal Soc. Bri. 97, Parts I-III, 438–475, Parts VII-VIII, 155–176 (1924)
9. Hill, D.W.: The critical power concept. Sports Medicine 16, 237–254 (1993)
10. Jones, A.M., Wilkerson, D.P., DiMenna, F., Fulford, J., Poole, D.C.: Muscle metabolic responses to exercise above and below the 'critical power' assessed using 31P-MRS. American Journal of Physiology: Regulatory, Integrative and Comparative Physiology 294, 585–593 (2008)
11. Lambert, E.V., St. Clair Gibson, A., Noakes, T.D.: Complex systems model of fatigue: integrative homoeostatic control of peripheral physiological systems during exercise in humans. British J. of Sports Medicine 39, 52–62 (2005)
12. Noakes, T.D.: Physiological models to understand exercise fatigue and the adaptations that predict or enhance athletic performance. Scand. Journal of Medicine and Science in Sports 10, 123–145 (2000)
13. Steels, L., Brooks, R.: The artificial life route to artificial intelligence: building embodied, situated agents. Lawrence Erlbaum, Mahwah (1995)
14. Wilmore, J.H., Costill, D.L.: Physiology of Sport and Exercise. Human Kinetics, Champaign (2004)
15. Widmaier, E.P., Raff, H., Strang, K.T.: Vander's Human Physiology: the Mechanisms of Body Functioning. MacGraw Hill (2006)

Appendix A: Some Further Details

This Appendix gives proof sketches for the lemma and propositions.

Proof of Lemma 1. Multiplying

$$R(t_2) = R(t_1) + \alpha_1 \int_{t1}^{t2} RP(t)dt - \alpha_2 \int_{t1}^{t2} EP(t)dt$$
$$DCP(t_2) = DCP(t_1) + \gamma_1 \int_{t1}^{t2} RP(t)dt - \gamma_2 \int_{t1}^{t2} EP(t)dt$$

by γ_1 respectively α_1 provides:

$$\gamma_1 R(t_2) = \gamma_1 R(t_1) + \gamma_1\alpha_1 \int_{t1}^{t2} RP(t)dt - \gamma_1\alpha_2 \int_{t1}^{t2} EP(t)dt$$
$$\alpha_1 DCP(t_2) = \alpha_1 DCP(t_1) + \alpha_1\gamma_1 \int_{t1}^{t2} RP(t)dt - \alpha_1\gamma_2 \int_{t1}^{t2} EP(t)dt$$

From subtracting, the statement of Lemma 1 follows. ■

Proof of Proposition 1

(i) ⇒ (ii) Suppose for two time points t_1 and t_2 it holds:

$$R(t_2) = R(t_1) = BR \text{ and } DCP(t_2) = DCP(t_1) = BCP$$

whereas between these time points resources have been extracted, after which full recovery took place, i.e.,

$$\int_{t1}^{t2} EP(t)dt \neq 0$$

By applying Lemma 1 it follows that $\alpha_1\gamma_2 - \gamma_1\alpha_2 = 0.$

(ii) ⇒ (i) This follows from Lemma 1, applied to the time interval between initial time point t_0 and t.

(ii) ⇔ (iii) ⇔ (iv) ⇔ (v) This also follows from Lemma 1. ■

Proof of Proposition 2

(i) ⇒ (iii) Apply Proposition 1 (iii) to a time interval from the initial time point t_0 and a time point t, to obtain that the expression $\alpha_1\ DCP(t) - \gamma_1\ \ R(t)$ is invariant over time. From this it follows that for all t

$$\alpha_1\ DCP(t) - \gamma_1\ \ R(t)\ = c$$

with

$$c = \ \alpha_1\ DCP(t_0) - \gamma_1\ \ R(t_0)\ =\ \alpha_1\ BCP - \gamma_1\ BR$$

From this it follows that

$$[\forall t\ \ DCP(t) = LCP\ \Leftrightarrow\ R(t) = 0\,]\ \Leftrightarrow\ c = \alpha_1\ LCP\ \Leftrightarrow \alpha_1\ (BCP\text{-}LCP) =\ \gamma_1\ BR$$
$$\Leftrightarrow\ \gamma_1/\alpha_1\ = \eta\ \Leftrightarrow\ \forall t\ \ \ DCP(t) = \eta\, R(t)$$

So under these conditions the dynamic critical power and the resource level are proportional over time with factor $\gamma_1/\alpha_1\ =\ \gamma_2/\alpha_2\ =\ \eta$. This proves Proposition 2. ■

A Model for Criminal Decision Making Based on Hypothetical Reasoning about the Future

Tibor Bosse and Charlotte Gerritsen

Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081, NL-1081 HV, Amsterdam, The Netherlands
{tbosse,cg}@few.vu.nl
http://www.few.vu.nl/~{tbosse,cg}

Abstract. This paper presents an agent-based model for decision making, which integrates personal biological and psychological aspects with rational utility-based reasoning. The model takes a BDI-based approach, where generation of desires is based on the personal characteristics, and generation of intentions is based on the rational reasoning. Moreover, a hypothetical reasoning mechanism is exploited to derive knowledge that connects certain actions to desires. The model has been implemented in the LEADSTO environment, and has been applied in a case study in the domain of criminal behaviour. Simulation experiments pointed out that the model enables agents to reason effectively about the consequences of their actions, which helps them to make the decisions that best satisfy their desires.

1 Introduction

Decision making is a complex process, which has received a lot of attention within various disciplines, ranging from psychology e.g., [1, 11, 16] and economics [17, 26] to computer science and artificial intelligence [12, 28]. Both for humans and artificial agents, the question 'which factors influence a certain decision' is important and non-trivial. A known problem encountered by modellers of decision making is that this process is determined partly by *rational* means-end reasoning, and partly by subjective personal *biological* and *psychological* aspects (including, for example, a person's motivational and emotional state), see, e.g., [9, 10, 20]. On the one hand, humans have various kinds of – partly biologically determined – desires, but on the other hand, they may have to reason rationally about which desires to fulfil. For example, a person may have the desire to hit someone but may decide rationally not to do this because this may have negative consequences. However, if the biological desire is too strong, the person may decide to hit nevertheless. Thus, humans exploit some mechanism that enables them to make decisions in situations where both rational and biological/psychological factors play a role.

This paper is part of a larger project, which has as main objective to develop a formal agent-based model of such decision making mechanisms. In principle, a generic approach will be taken, i.e., enabling modellers to formalise decision making in any arbitrary domain. More specifically, however, the current paper will focus on the

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 24–35, 2009.

domain of *crime*. This is an interesting case study, since this is a typical domain in which both rational decision making and biological and psychological aspects play a role. Within the area of criminology, a longstanding debate is whether criminal behaviour is driven by a criminal's personal biological background, or is the result of a rational, calculated choice; e.g., [8, 23]. The current paper will show how the two viewpoints can be integrated, thereby creating a behavioural model for a "criminal agent". Such a model can be useful from two perspectives. First, from a theoretical point of view, it may be used to get more insight in the process of criminal decision making itself. Second, from an application point of view, it may be used to develop *virtual agents* cf. [25] that are able to show criminal behaviour. Typical applications in which such criminal agents could play a role are (serious) games and computer-generated virtual stories e.g., [7].

As a starting point, the model described in [3] will be taken. This model addresses criminal action generation based on beliefs, desires and intentions (BDI), where generation of *desires* is based on biological and psychological aspects (such as high levels of testosterone or serotonin), and generation of *intentions* is based on rational, utility-based multi-criteria decision making cf. [17, 26]. According to this model, in case an agent has, for example, a high level of testosterone, this agent will generate a high desire for aggressive actions. When combining this desire with the belief that a certain action A will satisfy the desire, a state is generated in which A is considered a possible action. Next, this state is compared with other alternatives during a multi-criteria decision making process. However, one drawback of the model in [3] is that the knowledge that 'a certain action A satisfies a desire D' is assumed given. In contrast, in real world situations, humans do not always have such knowledge directly available. Instead, they often need to perform some kind of *hypothetical reasoning* [15, 30], in order to determine what will be the consequence of a certain (potential) action. Such reasoning may involve some derivation of possible effects in the world of a certain action, but also of effects on other persons (social cognition). For example, a criminal that is deliberating about whether or not to rob an old lady may take into account certain positive effects on the world (e.g., he will gain some money) and on other persons (e.g., his friends will admire him), as well as negative effects on the world (e.g., he might get caught) and on other persons (e.g., his parents will be disappointed). This paper extends the model from [3] with a model to determine such potential consequences of hypothetical actions.

Section 2 provides an overview of the literature used as a basis for the proposed model. In Section 3, the decision making model is presented, and Section 4 presents some simulation results. Section 5 provides a discussion about the work.

2 Hypothetical Reasoning, Social Cognition, and Rational Choice

Hypothetical reasoning[1] (also sometimes called 'what-if reasoning', or 'reasoning by assumption') is a practical reasoning pattern that often occurs within everyday life [15, 30]. The circumstances in which the pattern may be applied vary from diagnostic

[1] Note that this paper only addresses hypothetical reasoning about the future, in a way that is similar to temporal projection [13]. As such, it is different from hypothetical reasoning about the past, as is done in abduction [19].

problems to cases where persons deliberate about which action to perform. An example of the latter would be the following reasoning process: "*Suppose I do not take my umbrella with me. Then I run the risk of getting wet, which I don't want. Therefore I better take my umbrella with me*". The basic inference rule underlying this pattern is modus ponens: if p and p → q, then q. However, this example also involves some evaluation of the derived state q: since this state is in conflict with the person's desire, the hypothetical action p (not taking an umbrella) is rejected as being useful.

In the case of criminal behaviour, hypothetical reasoning may play an important role. Various criminologists describe criminals as rational agents that perform careful deliberation of alternatives before they decide to commit a crime e.g., [8]. During such deliberation, they may derive what is the effect of hypothetical actions on the world as well as on other agents. Concerning the latter, the influence of the peers can be very important. Especially adolescents are often put up to a certain action due to peer pressure. For them, it may be very important to belong to a group and have the respect of that group. A theory that is relevant while studying this phenomenon is the theory of *social cognition* by Mead [22]. According to Mead, social cognition arises in situations where people deliberate about planned actions that have impact on other people. Social cognition is a process that is similar to reciprocal role-taking between interactants, except that it occurs in the mind between two phases of the self, namely the *objective* and *acting* self. The objective self consists of a line of actions. This part of the self derives the possible consequences of the planned action and forms a view of the self performing this action from the standpoint of others. Next, the acting self reacts to this: it either responds positively to the objective self and allows the action to be performed or responds negatively and blocks it [21, 22].

In this paper, the ideas of hypothetical reasoning and social cognition are formalised, thereby creating a mechanism that evaluates the different aspects of a certain hypothetical action. These aspects may involve factors that have a biological background (e.g., the desire for aggressiveness) as well as factors that have a more rational nature (e.g., the desire to be appreciated by friends). For example: "*Suppose I attack that person. Then this will satisfy my desire for aggressiveness and my desire to be appreciated by my friends. Therefore I decide to attack that person*".

In the next section, this mechanism will be integrated with [3]'s model for decision making. This model is based on the *rational choice* theory within criminology; e.g. [8], which describes crime as the result of a deliberation process in which pros and cons are weighed. For example, an offender may decide to risk breaking the law, after assessing the chances of getting caught, the expected penalty, the value to be gained by committing the act, his or her immediate need for that value and the appreciation by his or her friends.

3 Simulation Model

In order to develop the model for criminal decision making, the modelling language LEADSTO [5] has been used. This language integrates qualitative, logical aspects and quantitative, numerical aspects, which allows the modeller to exploit both logical and numerical methods for analysis and simulation. Since the domain under consideration involves both qualitative aspects (e.g. decisions to perform a certain action) and quantitative aspects (e.g., utilities of a certain action), it is an appropriate modelling choice. However, in this paper LEADSTO is mainly used as a modelling vehicle. It is not claimed that it is the only possible approach[2].

In LEADSTO, direct temporal dependencies between two state properties in successive states are modelled by *executable dynamic properties.* The LEADSTO format is defined as follows. Let α and β be state properties of the form 'conjunction of ground atoms or negations of ground atoms'. In the leads to language the notation $\alpha \twoheadrightarrow_{e, f, g, h} \beta$, means:

> *If state property* α *holds for a certain time interval with duration g,*
> *then after some delay (between e and f) state property* β *will hold*
> *for a certain time interval of length h.*

Here, atomic state properties can have a qualitative, logical format, such as an expression desire(a, d), expressing that agent a has desire d, or a quantitative, numerical format such as an expression has_value(x, v), expressing that variable x has value v.

As mentioned in the introduction, the basis of the decision making model is a BDI-model, cf. [12, 28]. In this model an action is performed when the subject has the intention to do this action and it has the belief that the opportunity to do the action is there. Beliefs are created on the basis of stimuli that are observed. The intention to do a specific type of action is created if there is a certain desire, and there is the belief that in the given world state, performing this action will fulfil this desire. The BDI-model was specified by LEADSTO rule R1 and R2:

R1. Desire d combined with the belief that a certain action ac will lead to the fulfillment of that desire will lead to the intention to perform that action.

∀a:AGENT ∀d:DESIRE ∀ac:ACTION
desire(a, d) ∧ belief(a, satisfies(ac, d)) $\twoheadrightarrow_{0, 0, 1, 1}$ intention(a, ac)

In the presented model for criminal behaviour, desires may be complex states, which are composed of multiple sub-desires. These sub-desires have been determined based on interviews with domain experts. In the case study presented below, a desire of a criminal is composed of sub-desires for the following aspects: high gain, low loss, negative feelings, actions with strong stimuli, aggressiveness, and appreciation by friends (see [3] for details of the selected aspects). Such a composed desire is represented below as d(hg, ll, ne, ass, das, af). In addition, the following rule was used:

R2. The belief that there is an opportunity to perform a certain action combined with the intention to perform that action will lead to the performance of that action.

∀a:AGENT ∀ac:ACTION
belief(a, opportunity_for(ac)) ∧ intention(a, ac) $\twoheadrightarrow_{0, 0, 1, 1}$ performed(a, ac)

However, to assess and compare different options, and select a best option, as an extension to this basic BDI-model, personal *utilities* are to be assigned and combined, addressing the degree to which an action satisfies a desire of an individual. Note that these utilities are assumed to be subjective and personal. For example, for a criminal subject, due to his or her specific biological and psychological characteristics, a desire

[2] For example, other modelling languages that have similarities with LEADSTO (and its super-language TTL) are Situation Calculus [29], and Event Calculus [18]. For a comparison with these approaches, see [4].

may be quite deviant from what is commonly considered as the rational norm. For this subject, the utility of a certain action A is assessed according to the extent to which it fulfils this personal desire. This shows how utilities are assessed with respect to a subjective measure focusing on a specific desire D, which is affected, or even largely determined by the subject's specific biological and psychological background. According to this perspective, the utility-based decision model was set up as follows:

1. Aspect Utility Value Representations
For any aspect x_i, introduce an aspect utility v_i for any possible action ac by

has_aspect_utility(ac, x_1, v_1)
…
has_aspect_utility(ac, x_k, v_k)

where v_i expresses the expected utility for each aspect x_i of the considered action ac, normalised between 0 (minimal utility) and 1 (maximal utility). For example,

has_aspect_utility(fight, desire_for_aggressiveness_satisfied, 0.9)

indicates that the action of fighting contributes much to satisfaction of the desire for aggressiveness.

2. Aspect Weight Factor Representations
Introduce weight factors w_1, ..., w_k for the different aspects x_i, normalised so that the sum is 1, and introduce relations weight_factor(x_i, w_i) stating that aspect x_i has weight factor w_i.

3. Combination of Aspect Utilities to Option Utilities
Combine the option aspect utility values v_1, ..., v_k for a given composed desire to an overall option utility taking into account the weight factors w_1, ..., w_k, according to some combination function $f(v_1, \ldots, v_k, w_1, \ldots, w_k)$.

The combination function in **3.** can be formalised in a number of manners; two common possibilities are:

- Euclidian Distance**:** $f(v_1, \ldots, v_k, w_1, \ldots, w_k) = \sqrt{}(w_1v_1^2 + \ldots + w_kv_k^2)$
- Manhattan Distance**:** $f(v_1, \ldots, v_k, w_1, \ldots, w_k) = w_1v_1 + \ldots + w_kv_k$

The LEADSTO property for combination is:

R3. ∀a:AGENT ∀ac:ACTION ∀x_1,...,x_k:ASPECT ∀v_1,...v_k,w_1,...w_k:REAL
belief(a, has_aspect_utility(ac, x_1, v_1)) ∧ … ∧ belief(a, has_aspect_utility(ac, x_k, v_k)) ∧
weight_factor(x_1, w_1) ∧ … ∧ weight_factor(x_k, w_k) $\twoheadrightarrow_{0,\,0,\,1,\,1}$
belief(a, has_utility(ac, d(x_1,..., x_k)), $f(v_1, \ldots, v_k, w_1, \ldots, w_k)$))

Next, the choice process is formalised. This is done in two steps. First, R1 is replaced by R1a, R1b, and R1c:

R1a. Desire d combined with the belief that a certain action ac will lead to the fulfillment of d with utility u (≥c) will lead to the consideration of a as a possible intention option.

∀a:AGENT ∀d:DESIRE ∀ac:ACTION ∀u:REAL
desire(a, d) ∧ belief(a, has_utility(ac, d, u) ∧ u≥c) $\twoheadrightarrow_{0.2,\,0.2,\,1,\,1}$ is_intention_option(a, ac, u)

Here c is a threshold value, for example 0.5. This is used to generate the options to be considered. To obtain only the intentions with highest utility, as a next phase, the selection process is modelled in two steps by:

R1b. If ac1 and ac2 are both intention options, but ac2 has a higher utility, then ac1 is ruled out by the agent as an intention option.

∀a:AGENT ∀ac1,ac2:ACTION ∀u1,u2:REAL
is_intention_option(a, ac1,u1) $\wedge$ is_intention_option(a, ac2,u2) $\wedge$ u1<u2 $\twoheadrightarrow_{0,\,0,\,1,\,1}$
ruled_out_intention_option(a, ac1, u1)

R1c. Eventually, an intention option that is not ruled out is selected as final intention.

∀a:AGENT ∀ac:ACTION ∀u:REAL is_intention_option(a, ac, u) $\wedge$
not ruled_out_intention_option(a, ac, u)
$\twoheadrightarrow_{0,\,0,\,1,\,1}$ intention(a, ac)

The model presented so far enables an agent to deliberate between possible alternative actions ac, based on its desire d. However, it assumes that the knowledge about the extent to which actions satisfy desires (i.e., state properties of the form belief(a, has_utility(ac, d, u))) is given. Therefore, the following mechanism is proposed to derive such relations dynamically. The reasoning starts by assuming that the agent performs a certain (hypothetical) action ac (assumed(a, ac)). Next, knowledge about dependencies between world states (state properties of the form belief(a, leads_to(ac, y, v))) is exploited to predict the certainty with which actions lead to world states:

R4. If an agent currently assumes ac, and believes that ac leads to x with certainty v, then it will predict that ac leads to x with certainty v.

∀a:AGENT ∀ac:ACTION ∀x:INFO_ELEMENT ∀v:REAL
assumed(a, ac) $\wedge$ belief(a, leads_to(ac, x, v)) $\twoheadrightarrow_{0,\,0,\,1,\,1}$ predicted(a, leads_to(ac, x, v))

R5. If an agent currently predicts that ac leads to x with certainty v, and believes that x leads to y with certainty w, then it will predict that ac leads to y with certainty v*w.

∀a:AGENT ∀ac:ACTION ∀x,y:INFO_ELEMENT ∀v,w:REAL
predicted(a, leads_to(ac, x, v)) $\wedge$ belief(a, leads_to(x, y, w))
$\twoheadrightarrow_{0,\,0,\,1,\,1}$ predicted(a, leads_to(ac, y, v*w))

Finally, in case a prediction is made of an aspect that is relevant for the agent (because it has a desire for that aspect), then the probability of that aspect is taken as aspect utility.

R6. Predictions about relevant aspects are used as aspect utilities.

∀a:AGENT ∀ac:ACTION ∀x:INFO_ELEMENT ∀v:REAL
predicted(a, leads_to(ac, x, v)) $\twoheadrightarrow_{0,\,0,\,1,\,1}$ belief(a, has_aspect_utility(ac, x, v))

The complete utility-based decision model is depicted graphically in Fig. 1 (where the variables denoting the reasoning agent a have been left out, for simplicity). The circles denote state properties, and the arrows denote dynamic (LEADSTO) properties. Notice that the state properties of the type desire(...) can be generated by detailed submodels with domain-specific knowledge about particular types of criminals (e.g., as presented in [2]), which are outside the scope of this paper.

4 Example Simulation Trace

Based on the model shown above, a number of simulation experiments have been performed to test (for a simple scenario) whether it shows the expected behaviour. In this section, an example simulation trace is described in detail.

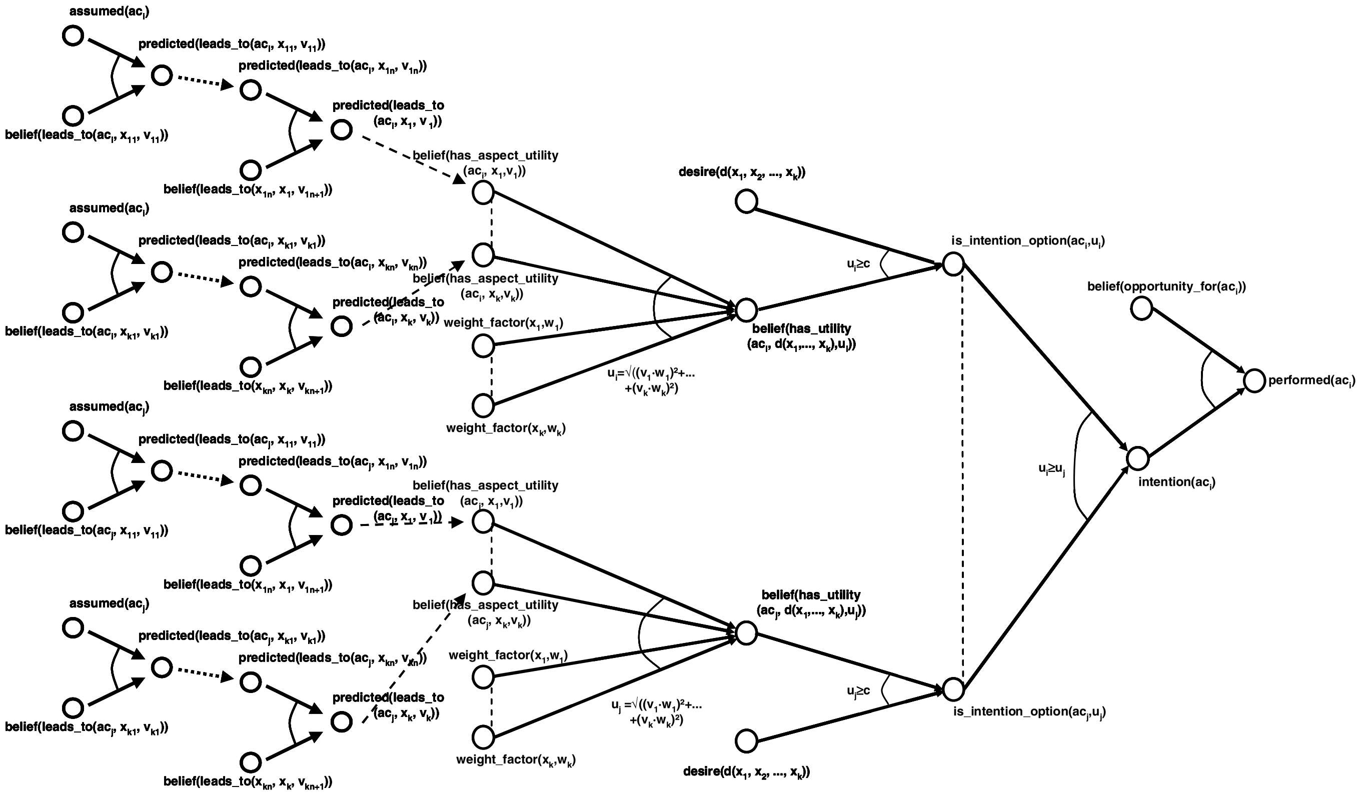

assumed(ac_i)
predicted(leads_to(ac_i, x_{11}, v_{11}))
belief(leads_to(ac_i, x_{11}, v_{11}))
predicted(leads_to(ac_i, x_{1n}, v_{1n}))
belief(leads_to(x_{1n}, x_1, v_{1n+1}))
predicted(leads_to (ac_i, x_1, v_1))
belief(has_aspect_utility (ac_i, x_1,v_1))
assumed(ac_i)
predicted(leads_to(ac_i, x_{k1}, v_{k1}))
belief(leads_to(ac_i, x_{k1}, v_{k1}))
predicted(leads_to(ac_i, x_{kn}, v_{kn}))
belief(leads_to(x_{kn}, x_k, v_{kn+1}))
predicted(leads_to (ac_i, x_k, v_k))
belief(has_aspect_utility (ac_i, x_k,v_k))
weight_factor(x_1,w_1)
weight_factor(x_k,w_k)
u_i=√((v_1·w_1)²+... +(v_k·w_k)²)
belief(has_utility (ac_i, d(x_1,..., x_k),u_i))
desire(d(x_1, x_2, ..., x_k))
u_i≥c
is_intention_option(ac_i,u_i)
assumed(ac_j)
predicted(leads_to(ac_j, x_{11}, v_{11}))
belief(leads_to(ac_j, x_{11}, v_{11}))
predicted(leads_to(ac_j, x_{1n}, v_{1n}))
belief(leads_to(x_{1n}, x_1, v_{1n+1}))
predicted(leads_to (ac_j, x_1, v_1))
belief(has_aspect_utility (ac_j, x_1,v_1))
assumed(ac_j)
predicted(leads_to(ac_j, x_{k1}, v_{k1}))
belief(leads_to(ac_j, x_{k1}, v_{k1}))
predicted(leads_to(ac_j, x_{kn}, v_{kn}))
belief(leads_to(x_{kn}, x_k, v_{kn+1}))
predicted(leads_to (ac_j, x_k, v_k))
belief(has_aspect_utility (ac_j, x_k,v_k))
weight_factor(x_1,w_1)
weight_factor(x_k,w_k)
u_j =√((v_1·w_1)²+... +(v_k·w_k)²)
belief(has_utility (ac_j, d(x_1,..., x_k),u_j))
desire(d(x_1, x_2, ..., x_k))
u_j≥c
is_intention_option(ac_j,u_j)
u_i≥u_j
intention(ac_i)
belief(opportunity_for(ac_i))
performed(ac_i)

Fig. 1. Overview of the Decision Making Model

The example scenario involves a boy A, who hangs out with his friends at school. These friends have the reputation of being bad boys. They like to harass their fellow students and make trouble. The friends have a negative influence on A, who actually is quite a nice boy. To be accepted a bit more by his friends he wants to impress them. An opportunity occurs; one of their fellow students arrives on his bike. Boy A thinks that attacking the student and taking his bike might have a good impact on the appreciation by his friends. Therefore he starts weighing the pros and cons. Boy A's individual weight factors for the different aspects involved are as follows: {high_gain = 0.14, low_loss = 0.17, negative_feelings = 0.08, actions_with_strong_stimuli = 0.19, desire_for_aggressiveness_satisfied = 0.17, appreciation_by_friends = 0.25}.

These weight factors, which add up to one, show the relative importance of each aspect for A. For example, the weight factor of high gain is 0.14, i.e., A does not find high gain very important. Appreciation, however, has a weight factor of 0.25: A finds it really important that his friends appreciate his behaviour. A is deliberating whether or not to perform the action (so to attack the student or do nothing).

Table 1 presents A's beliefs about dependencies between different world states. For each dependency, the first column shows the antecedent, the second column shows the consequent, and the third column shows the certainty A has about the dependency. For example, the fourth row is formalised via the state property:

```
belief(criminal,
      leads_to(belief(friend, performed(criminal, attack_student)), appreciation_by_friend, 0.9))
```

which expresses that the criminal believes that, when his friend knows that he attacked the student, he will probably appreciate this (with a probability of 90%).

Table 1. Agent A's Beliefs about World State Dependencies

antecedent	consequent	certainty
performed(criminal, attack_student)	world_state(performed(criminal, attack_student)	0.95
world_state(performed(criminal, attack_student)	observes(friend, performed(criminal, attack_student)	0.9
observes(friend,performed(criminal, attack_student)	belief(friend, performed(criminal, attack_student)	1.0
belief(friend,performed(criminal, attack_student)	appreciation_by_friend	0.9
world_state(performed(criminal, attack_student)	world_state(stolen(criminal, bike))	0.9
world_state(performed(criminal, attack_student)	world_state(performed(criminal, not_get_caught))	0.7
world_state(performed(criminal, attack_student)	experiences(student, suffering)	0.9
world_state(performed(criminal, attack_student)	experiences(criminal, adrenalin_rush)	1.0
world_state(performed(criminal, attack_student)	world_state(performed(criminal, hit_student)	0.8
performed(criminal, nothing)	world_state(performed(criminal, nothing)	0.95
world_state(performed(criminal, hit_student)	desire_for_aggresiveness_satisfied	0.9
world_state(stolen(criminal, bike))	high_gain	0.3
world_state(performed(criminal, not_get_caught))	low_loss	0.3
experiences(student, suffering)	negative_emotions	0.8
experiences(criminal, adrenalin_rush)	actions_with_strong_stimuli	0.6

Fig. 2 shows (part of) an example simulation trace that was generated on the basis of these settings. The example trace is the output of the LEADSTO simulation environment mentioned above [5]. Here, time is on the horizontal axis and state properties are on the vertical axis. A dark line indicates that a state property is true.

As can be seen in Fig. 2, agent A (represented as criminal in this simulation trace) has the opportunity to attack the student (belief(opportunity_for(performed(criminal, attack_student)))) or to do nothing (belief(opportunity_for(performed(criminal, do_nothing)))) at the beginning of the simulation. At time point 5, the criminal starts his deliberation process by assuming that he attacks the student. Then, he derives that this will lead to a probability of 85.5% that his friends will see this action (time point 7). He also derives that the attack of the student will probably lead to a stolen bike (85.5%), not getting caught (66.5%), suffering by the student (85.5%), an adrenalin rush (95%) and the student getting hit (76%). One step further, this might (but is not very certain to) lead to high gain (25.6%) and low loss (19.9%). It will probably lead to negative emotions (68%), the fulfillment of an action with strong stimuli (57%) and a satisfied desire for aggressiveness (68%), see time point 8. Furthermore, he derives that his friends will appreciate the action with 77% probability. The combination of these aspects makes that the student concludes that the attack is a possible action (is_intention_option) with a utility of 0.54. After that, A starts to consider the option of doing nothing. This decision making process is similar to the process of calculating the possible action of attacking the student (and was therefore left out of Fig. 2). Based on this process, the student concludes that doing nothing has a utility of 0.10. Since the action of attacking the student has a higher utility, the criminal decides to attack the student (at time point 22) and actually attacks the student (at time point 23).

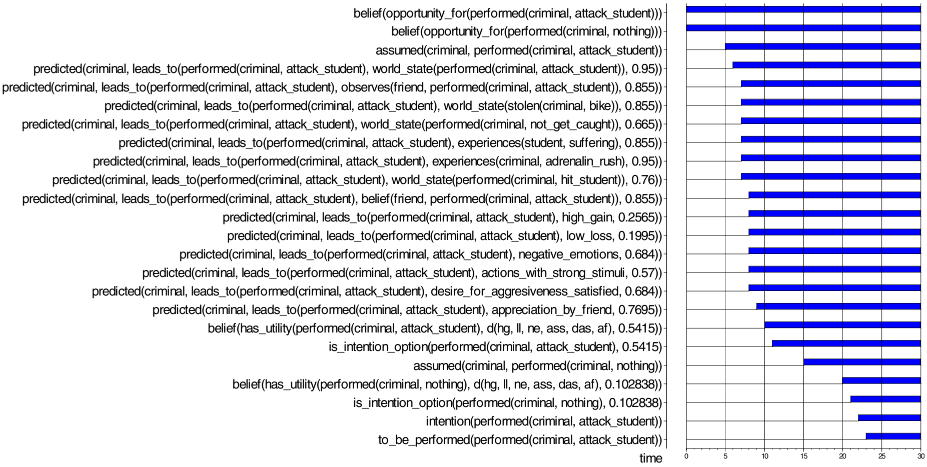


Fig. 2. Example Simulation Trace

All in all, this and similar scenarios have been simulated various times, using different parameters for the weight factors and the certainty of the dependencies involved. The computational time to generate such simulation traces was not more than a couple of seconds per trace. Due to space limitations, the details of these simulations are not shown here. Nevertheless, this case study indicated that the model enables agents to deliberate about the consequences of their actions, thereby weighing their pros and cons in the context of their (biologically and psychologically determined) desires, and eventually making a rational choice. It also shows that agents may make different decisions, based on personal characteristics and priorities. The resulting simulation traces have been shown to expert criminologists, who confirmed the plausibility of the produced patterns, and their correspondence to literature such as [8, 23]. Although this is not an exhaustive validation, it is an indication that the model produces expected behaviour.

5 Discussion

This paper presented an agent-based model for decision making, which integrates personal biological and psychological aspects (e.g. high levels of testosterone) with rational utility-based reasoning. The model takes a BDI-based approach, where generation of desires is based on the personal characteristics, and generation of intentions is based on the rational reasoning. Compared to an earlier version [3], the current paper specifically adds a mechanism to derive knowledge that connects certain actions to desires. This is done by means of hypothetical reasoning: the agent imagines performing an action and applies a forward reasoning strategy in order to derive the consequences, and to what extent these fulfil the agent's desires.

The model has been implemented in the LEADSTO environment [5]. For a case study in criminal behaviour, various simulation experiments have been performed. These simulations have pointed out that the model enables agents to reason about the consequences of their actions, and to take this into account during decision making.

The presented model was inspired by related approaches in the area of Artificial Intelligence. In recent years, several authors have proposed agent-based models for reasoning and decision making (which often also combine rational and non-rational aspects), such as in [14, 31]. A difference with the presented model is that most of these models explicitly focus on the integration of *emotions* with rational behaviour, whereas our model tries to integrate rational behaviour with personal biological and psychological factors *in general* (including emotions, but also notions like aggressiveness and arousal). Furthermore, the presented model has some similarities with standard BDI-based agent modelling frameworks, such as AgentSpeak [27] or Jadex [24]. In further research, it may be investigated whether the proposed decision making model can be implemented in such frameworks.

Concerning further future work, these are several possibilities. As mentioned in the introduction, the presented model can be useful from two perspectives. First, it may be used to get more insight in the process of criminal decision making itself: why do certain people commit certain crimes? To answer such questions, the model has to be fed with more detailed domain knowledge from criminological and psychological literature, and specific experiments have to be designed to study typical questions described in the literature. Some initial steps in this direction have been made, see also [2, 3]. Second, the model can be used to develop *virtual agents* cf. [25] that are able to make decisions based on hypothetical reasoning, and more specifically, to develop 'criminal agents'. For example, such criminal agents may play a role in (serious) games and computer-generated virtual stories e.g., [7]. Since the LEADSTO language has an intuitive (causal relationship-based) format, and is independent of a particular implementation language, this step is relatively straightforward, see also [6]. When doing that, also LEADSTO's possibilities to specify non-deterministic reasoning rules can be exploited. Work is currently in progress to elaborate these ideas in detail.

References

1. Baron, J.: Thinking and Deciding. Cambridge University Press, Cambridge (1988)
2. Bosse, T., Gerritsen, C., Treur, J.: Integration of biological, psychological, and social aspects in agent-based simulation of a violent psychopath. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2007. LNCS, vol. 4488, pp. 888–895. Springer, Heidelberg (2007)
3. Bosse, T., Gerritsen, C., Treur, J.: Integrating Rational Choice and Subjective Biological and Psychological Factors in Criminal Behaviour Models. In: Lewis, R.L., Polk, T.A., Laird, J.E. (eds.) Proceedings of the 8th International Conference on Cognitive Modeling, ICCM 2007, pp. 181–186. Taylor and Francis, Abington (2007)
4. Bosse, T., Jonker, C.M., van der Meij, L., Sharpanskykh, A., Treur, J.: Specification and Verification of Dynamics in Agent Models. Int. Journal of Cooperative Information Systems (2009)
5. Bosse, T., Jonker, C.M., van der Meij, L., Treur, J.: A Language and Environment for Analysis of Dynamics by Simulation. Int. Journal of AI Tools 16(3), 435–464 (2007)
6. Bosse, T., de Lange, F.P.J.: Development of Virtual Agents with a Theory of Emotion Regulation. In: Proc. of the Eighth IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2008. IEEE Computer Society Press, Los Alamitos (2008) (in press)
7. Cavazza, M., Charles, F., Mead, S.: Interacting with virtual characters in interactive storytelling. In: Alonso, E., Kudenko, D., Kazakov, D. (eds.) AAMAS 2000 and AAMAS 2002. LNCS (LNAI), vol. 2636, pp. 318–325. Springer, Heidelberg (2003)
8. Cornish, D.B., Clarke, R.V.: The Reasoning Criminal: Rational Choice Perspectives on Offending. Springer, Heidelberg (1986)
9. Damasio, A.R.: Descartes' error: Emotion, reason, and the human brain. Putnam, NY (1994)
10. Evans, J.S.B.T.: In two minds: dual-process accounts of reasoning. Trends in Cognitive Sciences 7, 454–459 (2003)
11. Fishburn, P.C.: Utility Theory for Decision Making. John Wiley and Sons, NY (1970)
12. Georgeff, M.P., Lansky, A.L.: Reactive Reasoning and Planning. In: Proc. of the Sixth National Conf. on Artificial Intelligence, AAAI 1987, Menlo Park, pp. 677–682. American Association for Artificial Intelligence (1987)
13. Hanks, S., McDermott, D.: Nonmonotonic logics and temporal projection. Artificial Intelligence 33, 379–412 (1987)
14. Jiang, H., Vidal, J.M., Huhns, M.N.: EBDI: An Architecture for Emotional Agents. In: Proc. of the 6th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems, AAMAS 2007, pp. 38–40. ACM Press, New York (2007)
15. Jonker, C.M., Treur, J.: Modelling the Dynamics of Reasoning Processes: Reasoning by Assumption. Cognitive Systems Research 4, 119–136 (2003)
16. Kahneman, D., Tversky, A.: Prospect theory: An analysis of decision under risk. Econometrica 47, 263–291 (1979)
17. Keeney, R., Raiffa, H.: Decisions with Multiple Objectives: Preferences and Value Trade-offs. John Wiley & Sons, Chichester (1976)
18. Kowalski, R., Sergot, M.: A logic-based calculus of events. New Generation Computing 4, 67–95 (1986)
19. Lifschitz, V.: On the logic of causal explanation. Artificial Intelligence 96(2), 451–465 (1997)

20. Loewenstein, G.F., Weber, E.U., Hsee, C.K., Welch, E.S.: Risk as feelings. Psychological Bulletin 127, 267–286 (2001)
21. Matsueda, R.L., Kreager, D.: Social Cognition, Identity, and Deterrence: a model of criminal decisions. In: Proc. of the Annual Meeting of the Am. Sociological Association (2004)
22. Mead, G.H.: In: Morris, C.W. (ed.) Mind, Self and Society: from the standpoint of a social behaviorist. University of Chicago, Chicago (1934)
23. Moir, A., Jessel, D.: A Mind to Crime: the controversial link between the mind and criminal behaviour. Michael Joseph Ltd.; Penguin, London (1995)
24. Pokahr, A., Braubach, L., Lamersdorf, W.: Jadex: A BDI reasoning engine. In: Bordini, R.H., Dastani, M., Dix, J., El Fallah Seghrouchni, A. (eds.) Multi-Agent Programming: Languages, Platforms and Applications, pp. 149–174. Springer, Heidelberg (2005)
25. Prendinger, H., Lester, J., Ishizuka, M. (eds.): IVA 2008. LNCS (LNAI), vol. 5208. Springer, Heidelberg (2008)
26. Raiffa, H.: The Art and Science of Negotiation. Harvard University Press, Cambridge (1982)
27. Rao, A.S.: AgentSpeak(L): BDI agents speak out in a logical computable language. In: Perram, J., Van de Velde, W. (eds.) MAAMAW 1996. LNCS (LNAI), vol. 1038, pp. 42–55. Springer, Heidelberg (1996)
28. Rao, A.S., Georgeff, M.P.: Modelling Rational Agents within a BDI-architecture. In: Allen, J., et al. (eds.) Proc. of the 2nd Int. Conference on Principles of Knowledge Representation and Reasoning (KR 1991), pp. 473–484. Morgan Kaufmann, San Francisco (1991)
29. Reiter, R.: Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems. MIT Press, Cambridge (2001)
30. Rescher, N.: Hypothetical reasoning. North-Holland, Amsterdam (1964)
31. Silverman, B.G.: More Realistic Human Behavior Models for Agents in Virtual Worlds: Emotion, Stress, and Value Ontologies. University of Penn./ACASA, T.R., Philadelphia (2001)

An Agent Model of Temporal Dynamics in Relapse and Recurrence in Depression

Azizi A. Aziz, Michel C.A. Klein, and Jan Treur

Agent Systems Research Group, Department of Artificial Intelligence
Vrije Universiteit Amsterdam, De Boelelaan 1081a,
1081 HV Amsterdam, The Netherlands
{mraaziz,michel.klein,treur}@few.vu.nl

Abstract. This paper presents a dynamic agent model of recurrences of a depression for an individual. Based on several personal characteristics and a representation of events (i.e. life events or daily hassles) the agent model can simulate whether a human agent that recovered from a depression will fall into a relapse or recurrence. A number of well-known relations between events and the course of depression are summarized from the literature and it is shown that the model exhibits those patterns. In addition, the agent model has been mathematically analyzed to find out which stable situations exist. Finally, it is pointed out how this model can be used in depression therapy, supported by a software agent.

Keywords: Agent Based Modeling, Temporal Dynamics, Relapse and Recurrent in Depression**.**

1 Introduction

Unipolar depression is a mental disorder distinguished by a persistent low mood, and loss of awareness or contentment in usual activities [2]. Despite the modern era of pharmaceutical and holistic intervention, one of the primary problems with unipolar depression (i.e. a depression not related to other mental disorders) is that it has a very high rate of recurrent and relapse cases [14]. At least 60 percent of individuals who have had one depressive episode will have another, 70 percent of individuals who have had two depressive episodes will have a third, and 90 percent of individuals with three episodes will have a fourth episode [1]. Although the risk of relapse may decline with time, even for those who remain well for 5 years after an index episode, the rate of recurrence/relapse is 58 percent [3]. Despite the magnitude of the problem of recurrence and relapse, little attention has been focused on the symptom pattern in recurrent episodes of major depression [1][14]. In practice, there is a need to have a mechanism to monitor the condition of individuals who have had a previous encounter with unipolar depression, eventually improving their quality of life. In order to achieve this objective, the aim of the embedding research project is to develop an agent-based application that is able to support humans in the long term. The software agent is expected to have capabilities to understand its environment and the individual, providing a better monitoring and assessment of the situation. To implement this capability in any software agent, it is required to incorporate a human

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 36–45, 2009.

agent model that shows how humans might fall into relapse / recurrence or stay healthy. In case a relapse or recurrence is predicted, the agent can provide to support by providing adequate remedies.

This paper focuses exclusively on the formal model for dynamics in relapse/ recurrence, as it is one of the essential components in the development of a software agent that is able to monitor individuals' conditions. In the next section, the underlying principles in relapse and recurrence in unipolar depression are discussed (Section 2). From this perspective, a formal model is designed and formulated (Section 3). Later, in Section 4, simulation traces are presented to illustrate how this model satisfies the expected outcomes in recurrent / relapse. In Section 5, a detailed mathematical analysis is performed, to identify equilibria in the model. Finally, Section 6 concludes the paper.

2 Underlying Principles of Relapse and Recurrence in Depression

Before presenting the model, the main characteristics of recurrence and relapse of depression as known from the literature are described. First, the effect of repeated stressful events is explained. Then, the knowledge about the causes of relapse and recurrence are discussed.

Frequent stressful events (stressors) are correlated with a positive contribution to the development of recurrence and relapse [3]. Contrary to popular belief, repeated strikes, even when they are low, can have almost the same effect as a similar single instantaneous stressful event [1][7]. This can be explained by an analogy of striking a bell. Imagine when a bell is struck once, it emits a sound that is loud at first, and then decays in intensity. However, if each subsequent strike is applied before the sound of the preceding strike has diminished: the loudness will increase each time. Applying this to the real world, a single and low stressor event may initially be so miniscule that it is considered to cause no effect. However, many repeated and small stressor events will eventually lead to a higher level of potential stress than a single major stress-producing event [3] [12]. Therefore, the intensity of a single stressor event faced by an individual is not the only important factor, because if negative events are persistently present, they can have a stronger effect than an initial event with a higher intensity.

A key step in the development of a model to represent potential onset of relapse and recurrence is to understand how this condition may recur [12]. Stressors from the environment are the dominant components that will lead to recurrence or relapse of depression [9]. This primary mechanism however is regulated by two main apparent predisposing factors, which influence the process as moderators that can neutralize each other. These two components are simplified as *immunity* and *neuroticism* (vulnerabilities in the personality) [3][5][6]. These factors are induced by the observed evidences that there are personal differences and conditions that will increase or decrease the onset of recurrence or relapse in any individual [9][11]. In addition, in many works, these two components are assumed to influence not only the possibility of onset of a depression, but also affect the duration of it [11][12]. On the other hand there are many factors that eventually help people to sustain their well-being. These factors are closely related to: (a) *coping skills*, (b) *being assertive*, and

(c) *knowing when to seek help* [4][6][8]. The first is the ability to cope with the adversities. *Coping skill* is a behavioral and biologically wired tool which may be used by individuals to offset stressor events without correcting or eliminating the underlying condition. On the basis of many theories in depression, coping responses and strategies have been most frequently divided into problem focused coping and avoidant coping responses [9][10]. *Problem focus coping* responses allow an individual to increase the perceived control over their problem; it is proven in many studies that they are able to reduce the risk of onset of a depression [4]. They involve attempts to do something constructive about the stressful conditions that are harming an individual, rather than to make it worsen. In contrast, *avoidance coping* is detrimental in nature. When feelings of discomfort appear through stressor events, people find ways of not experiencing them. Such a strategy may work in a very short term, but it is mostly considered as an inadequate approach of coping [8]. The second component is being *assertive*. Individuals who are assertive tend to be aware of their emotions and communicate these in clear-cut manner and are able to make and meet goals and challenges through respect and perseverance [5]. In many cases, people with a high assertive level are more likely to be more proactive and problem focused rather than unassertive individuals [1][9].

The last component is the ability to seek *social support*, ("having positive interaction of helpful behavior provided to a person in need of support") [4]. As a multidimensional concept in nature, social support also includes many other facets that may finally determine if social support is constructed such as having the ability to create a support network [8][13]. There are many characteristics of individuals that influence their potential abilities of seeking support or vice versa. For example, an individual who is highly neurotic, using more avoidant coping and having a lack of self esteem may not be able to request support, and later it may disengage him/herself from potential social support.

In short, the following relations can be identified from the literature: *(1)* a series of smaller stressor events can lead to the recurrence or relapse; *(2)* stressor events directly affect the potential onset of relapse /recurrence; *(3)* neuroticism aggravates the effect of stressor events on the potential onset of a depression;*(4)* assertiveness and immunity will diminish the potential of onset, and *(5)* a combination of good social support and coping skills will reduce the risk of having future relapse/ recurrence.

3 Model for Relapse and Recurrence

The characteristics of the proposed model are heavily inspired by the research discussed in the previous section on recurrence and relapse, especially in depression. In particular, this model combines ideas from research in affective disorder, prevention medicine, artificial intelligence, and dynamic modeling. Those ideas are encapsulated in a way that allows to simulate how an individual is fragile towards stressors, and possibly further develops a depression. All of these concepts (and their interactions) are discussed in the following paragraphs in this section.

3.1 Formalizing the Model Relationships

In this model, there are four major components that will represent dynamic interactions of human agent abilities involved in recurrence/relapse namely; *environment*, *personality*, *social support*, and *coping strategies*. By combining these characteristics together, it will allow a hypothesis or expected behavior for the human agent to be monitored.

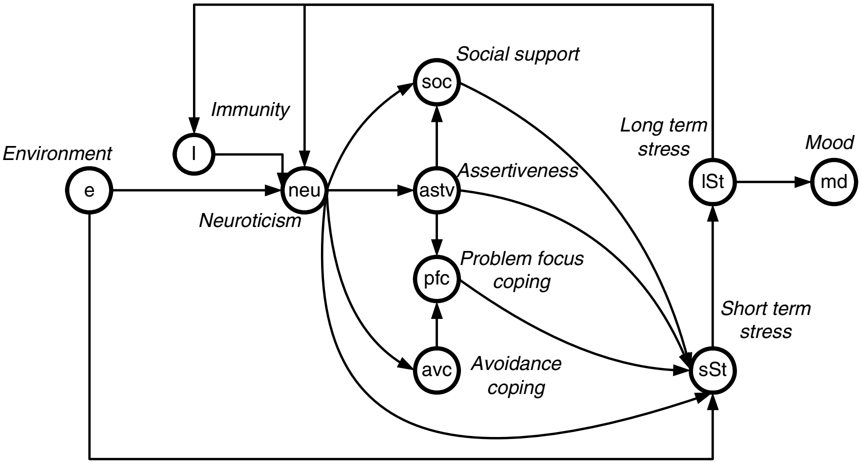


Fig. 1. Global Relationships of Variables Involved in the Onset of Relapse and Recurrence Depression

Once the structural relationships in the model have been determined, the model can be formalized. In the formalization, all nodes are designed in a way to have values ranging from 0 (low) to 1 (high). The interaction will determine the new value of it, either by a series of accumulations or an instantaneous interaction for each node.

Stressor Events: In the model, the stressor events *(e)* are generated by simulating potential effects throughout *t* time using weighted sum of three types of events; life (*le*), chronic (*ce*), and daily (*de*) events.

$$e(t) = w_1.le(t) + w_2.ce(t) + w_3.de(t), \tag{1}$$

The role of these factors in the model is to represent a series of events. Stressors are seen as very intense when $e(t) \rightarrow 1$, and no stressors are represented by $e(t) \rightarrow 0$.

Dynamics of Neuroticism: In this model, the neurotic level *(neu)* describes the interactions between environment (*e*), personal immunity trait (*I*), and prior exposure to long-term stress (*lSt*), in a time interval between *t* and *t*+Δ*t*. Here, α_{neu} is a parameter for a change rate, and β_{neu} is a parameter for the contribution effect of the previous neurotic rate in this equation.

$$neu(t+\Delta t) = neu(t) + \alpha_{neu.}.(1-neu(t)).[(f(e(t),I(t)).lSt(t))-\beta neu.neu(t))].\ \Delta t \tag{2}$$

where, $f(e(t),I(t))$ is a logistic unit function, $1/(1+\eta e^{-\alpha(e(t)-I(t))})$.

Social Support, Problem Focus Coping, Assertiveness, and Immunity: Social support (*soc)* is computed by multiplying the factor of being assertive by the ability of less or non- neurotic. Problem focus coping *(pfc)* is also computed with the same approach, but with a negative association in avoidant coping (*avc*)*.* The α_{avc} is a proportional rate for the effect of neurotic level in *avc*.

$$soc(t)=astv(t).(1-neu(t)) \tag{3}$$

$$pfc(t)=astv(t).(1-avc(t)) \tag{4}$$

$$avc(t)=\alpha_{avc}.neu(t) \tag{5}$$

Assertiveness *(astv)* depends on the interaction between the normal assertive value within an individual and the condition of being less or non-neurotic. The immunity *(I)* level interaction also having a similar behavior, but it is related to the interaction in a long term stress level.

$$astv(t) = \alpha_{astv}.astv_{norm}+(1-\alpha_{astv}).(1-neu(t)).\ astv_{norm} \tag{6}$$

$$I(t) = \alpha_{I}.I_{norm}+(1-\alpha_{I}).(1-lst(t)).\ I_{norm} \tag{7}$$

Dynamics of Short Term Stress, Long Term Stress, and Mood: Short term stress *(sSt)* is modeled by instantaneous relationships between the environment, neurotic level, and reducer components, ψ (a combination of social support, assertiveness, and problem focus coping). Long term stress *(lSt)* is primarily contributed the accumulation exposure towards short term stress and later will influence the level of mood *(md)* in a time interval between *t* and *t*+Δ*t*.

$$sSt(t)=\beta_{sst}.e(t)+(1-\beta_{sst}).neu(t).(1-e(t)).(1-\psi) \tag{8}$$

$$lSt(t+\Delta t) = lSt(t)+ \alpha_{lst}.\ (1- lSt(t)).(sSt(t)- \beta_{lst}\ lSt(t)).\ \Delta t \tag{9}$$

$$md(t+\Delta t) =md(t)+ \eta_{md}.\ (1-md(t)).(lSt(t)- \beta_{md.}md(t)).\ \Delta t \tag{10}$$

where η_{md} ,β_{md} ,α_{lSt} , β_{sSt} and β_{lSt} denote the proportion change rates for all respective equations.

4 Example Simulation Traces

In this section, the model was executed to simulate a large number of conditions of individuals. In this section, three examples are shown: a healthy individual **(A)**, an individual with a potential risk of relapse and recurrence **(B),** and an individual with severe conditions **(C).** The initial settings for the different individuals are the following (*neu (t=0), astv*$_{norm}$*, I*$_{norm}$); **A** (*0.1, 0.8, 0.8*), **B** (*0.5, 0.5, 0.4*), and **C** (*0.8, 0.2, 0.1*). In all cases, the initial long term stress and mood value is initialized as 0.3, 0.1 respectively. Corresponding to these settings, the level of severity (or potential onset) is measured, defining that any individuals scored more than 0.5 in their mood level (within more than 336 time steps) will be considered as reaching the recurrent or relapse stage. These simulations used the following parameters settings: t_{max}*=1000* (to represent a monitoring activity up to 42 days), Δ*t*=0.3*,* α_{lst}=0.3,

η_{md}=0.2, β_{sst}=0.3, α_{l}=0.7, α_{astv}=0.5, α_{avc} =0.5, α_{neu}=0.4, and with all decay terms are assigned as 0.02.

Result # 1: Simulation Trace for Prolonged-Fluctuating Stressor Events
During this simulation, each type of individual has been exposed to an extreme stream of stressor events, with a rapid alteration between each corresponding event. This kind of pattern is comparable to the repeated strike event, where stressor events always occur when the previous events were ended.

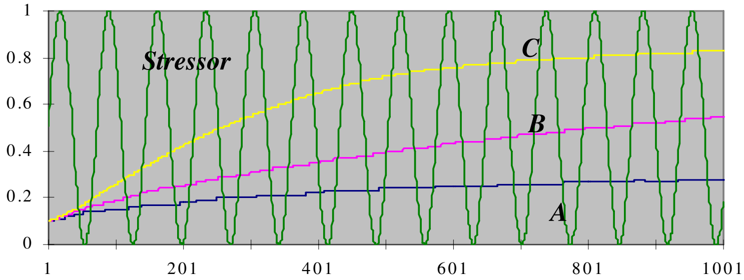


Fig. 2. Relapse / Recurrent Onset for Each Individual in Prolonged Stressor Events

In this simulation trace, it shown that an individual C (*high neurotic, low in assertive* and *immunity*) tends to get into onset much faster compared to other individuals. Note that the individual B (*moderate neurotic, assertive,* and *immunity*) shows a gradual increasing level of potential onset and possibly will experience relapse / recurrent if that individual is having constant exposure towards stressors. Individual A however is less prone to develop a potential onset condition within a short period of time.

Result # 2: Simulation Trace for Decrease Stressor Events
This simulation trace shows two types of periods, one with a very high constant and with a very low constant stressor event. These events occurred in a constant behavior for a certain period of time (approximately within 20 days).

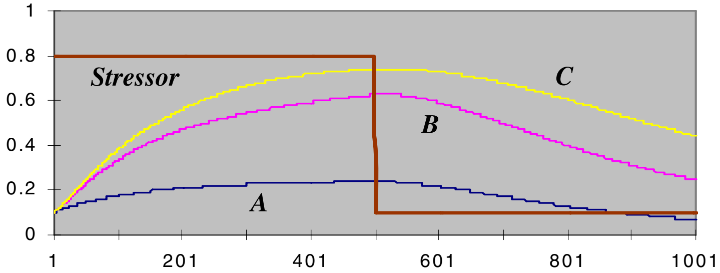


Fig. 3. Relapse / Recurrent Onset for Each Individual in Fluctuated Stressor Events

Also here it can be seen that individual C gets into a bad mood much faster than the others. Moreover, even at the end of the simulation time, the mood of individual C is worse than the mood of the other two individuals. Using a similar experimental setting, with t_{max}=10,000, the end of the experimental results show all individuals will have a normal mood level.

Result # 3: Simulation Trace with Social Support, and Problem Focus Coping Skills (Reducer)

As initially discussed in Section 2, a combination of social support, and problem focus coping skills is expected to help any individuals to reduce potentially risk in relapse / recurrence. The combination of these factors will be represented by R_A, $R_{B,}$ and R_c for the respective human agents. To visualize the effect of these, frequently repeating low stressor events were simulated. Figure 4 illustrates how these combinations, mood levels and stressor events are influencing each other.

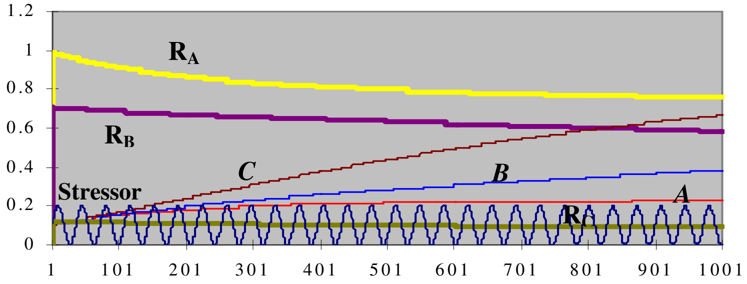


Fig. 4. Relapse / Recurrent Onset for Each Individual with a Combination of Reducer

Figure 4 shows that a healthy individual (*A*) has much higher reducer factors than less healthy individuals. These reducing factors limit the effect of the incoming stressors. Also it can be seen that the reducer factors decrease over time, due to the relatively low but frequent stressors. The patterns for the different individuals are the same as in Figure 2, although the final mood level is lower in Figure 4 because of the less intense stressors fluctuation.

To wrap up these experimental results, the simulation traces described above satisfactorily explain the relations as summarized in Section 2. In all simulation traces, it is shown that individuals with higher assertiveness, immunity and less neurotic levels develop less often a relapse compared those who are not. In addition, a higher neurotic level eventually aggravates the potential risk of onset, as illustrated in all simulation traces. The effects of stressor events on relapse/ recurrence onset are also exemplified. In all simulation traces, it is apparent that frequent or high stressor events contribute to the potential risk of onset. Furthermore, the effect of the reducers is also examined, where in Figure 4, it depicts that when the reducer level is decreasing, the person is also prone to a relapse, or vice versa. This distillation of above evidences and traces illustrates that this model reflects the basic relations that are known to influence relapse and recurrence, given certain criteria of events and personality attributes.

5 Mathematical Analysis

In this section the equilibria are analyzed that may occur under certain conditions. The equilibria describe situations in which a stable situation has been reached. Those equilibria are interesting as it should be possible to explain them using the knowledge of the domain that is modeled. As such, the existence of reasonable equilibria is an indication for the correctness of the model. To analyze the equilibria, the available temporal and instantaneous equations are filled with values for the model variables such that the derivatives or differences between time point t and $t + \Delta t$ are all 0 (in particular for neuroticism, long term stress and mood). Moreover, for an equilibrium, the external input is also assumed to be constant. To start, for an equilibrium for neuroticism it holds as ; $(1\text{-}neu).(f(e,I).lSt\ \text{-}\beta_{neu}.neu) = 0.$

This is equivalent to $neu = 1$ or $neu = f(e,I).lSt/\beta_{neu}$, where $f(e,I) = 1/(1+\eta e^{-\alpha(e-I)})$. Assuming high steepness of the threshold function provides the cases $e \leq I$ (where $f(e,I)= 0$) or $e > I$ (where $f(e,I)= 1$). Under this assumption the cases are $neu = 1$ or $neu = 0$ and $e \leq I$ or $neu = lSt/\beta_{neu}$ and $e > I$. For an equilibrium for assertiveness it holds;

$$\begin{aligned} astv &= \alpha_{astv}.astv_{norm} + (1\text{-}\alpha_{astv}).(1\text{-}neu).astv_{norm} \\ &= astv_{norm}\ \text{-}\ (1\text{-}\alpha_{astv}).\ neu.\ astv_{norm} \end{aligned}$$

Meanwhile, for an equilibrium for immunity it holds:

$$I = \alpha_I.I_{norm}+(1\text{-}\alpha_I).(1\text{-}lst).\ I_{norm} = I_{norm}\ \text{-}\ (1\text{-}\alpha_I).lSt.I_{norm}$$

For an equilibrium for long term stress it holds $(1\text{-}\ lSt).(sSt\text{-}\ \beta_{lst}\ lSt) = 0,$ which is equivalent to $lSt = 1$ *or* $sSt = \beta_{lst}\ lSt.$ For an equilibrium for mood it holds $(1\text{-}md).(lSt\text{-}\beta_{md.}md) = 0$ which is equivalent to $md = 1$ *or* $md = lSt\ /\beta_{md}.$ Table 1 provides a summarization of these equilibria.

Table 1. Equilibrium Equations for Respective Variables

Var.	Equilibrium equations
neu	$neu = 1$ or $e \leq I$ and $neu = 0$ or $e > I$ and $\beta_{neu}\ neu = lSt$
$astv$	$astv = astv_{norm}\ \text{-}\ (1\text{-}\alpha_{astv}).\ neu.\ astv_{norm}$
soc	$soc = astv.(1\text{-}neu) = (\ astv_{norm}\ \text{-}\ (1\text{-}\alpha_{astv}).\ neu.\ astv_{norm})(1\text{-}neu)$
avc	$avc = \alpha_{avc}.neu$
pfc	$pfc = astv.(1\text{-}avc) = (\ astv_{norm}\ \text{-}\ (1\text{-}\alpha_{astv}).\ neu.\ astv_{norm})\ (1\text{-}\alpha_{avc}.neu)$
I	$I = I_{norm}\ \text{-}\ (1\text{-}\alpha_I).lSt.\ I_{norm} = I_{norm}\ \text{-}\ (1\text{-}\alpha_I).lSt.\ I_{norm}$
sSt	$sSt = \beta_{sst}.e+(1\text{-}\beta_{sst}).neu.(1\text{-}e).(1\text{-}\psi)$ *where* $\psi = w_1.soc + w_2.pfc+w_3.astv$ $=\beta_{sst}.e+(1\text{-}\beta_{sst}).neu.(1\text{-}e).(1\text{-}(w_1.(astv_{norm}\text{-}(1\text{-}\alpha_{astv}).neu.$ $astv_{norm})(1\text{-}neu))+w_2(astv_{norm\text{-}}(1\text{-}\alpha_{astv}).neu.astv_{norm})(1\text{-}$ $\alpha_{avc}.neu))+w_3.(\ astv_{norm}\ \text{-}\ (1\text{-}\alpha_{astv}).\ neu.\ astv_{norm})))$
lSt	$lSt = 1$ or $sSt = \beta_{lst}\ .lSt$
md	$md = 1$ or $md = lSt\ /\ \beta_{md}$

It turns out that all values can be expressed in terms of either *neu* or *lSt*: in the Table 1 the values *astv, soc, avc, pfc* have been expressed in *neu*, and the values *md, sSt, I* have been expressed in *lSt.* Then by the equation for short term stress the value *lSt* can be expressed in *neu*.

$$\beta_{lst}\, lSt = \beta_{sst}.e+(1-\beta_{sst}).neu.(1-e).(1-(w_1.(astv_{norm} - (1-\alpha_{astv}).\, neu.\, astv_{norm})(1-neu)) \\ +w_2(astv_{norm} - (1-\alpha_{astv}).\, neu.\, astv_{norm})(1-\alpha_{avc}.neu))+w3.(astv_{norm} - (1- \\ \alpha_{astv}).\, neu.\, astv_{norm})))$$

From the equation for neuroticism two cases occur; $e \leq I$ and $neu = 0$ or $e > I$ and $\beta_{neu}\, neu = lSt$. These cases will be addressed in some more detail.

Analysis of Case e ≤ *I* and *neu = 0*: In this case, the following values are found:

$$neu = 0,\ astv = astv_{norm}\ ,\ soc = astv_{norm},\ avc = 0,\ pfc = astv_{norm} \\ sSt = \beta_{sst}.e,\ lSt = \beta_{sst}.e\,/\beta_{lst},\ I = I_{norm} - (1-\alpha_I).\ \beta_{sst}.e\,/\beta_{lst}.\ I_{norm} \\ md = 1 \quad \text{or} \quad md = \beta_{sst}.e\,/\beta_{lst}\,\beta_{md}$$

Here the condition $e \leq I$ is equivalent to: (1) $ie \leq I_{norm} - (1-\alpha_I).\ \beta_{sst}.e\,/\beta_{lst}.\ I_{norm}$,

(2) $e\,(1+(1-\alpha_I).\ \beta_{sst}.\,/\beta_{lst}.\ I_{norm}) \leq I_{norm}$, *and* (3) $e \leq I_{norm}\ /(1 +(1-\alpha_I).\ \beta_{sst}.\,/\beta_{lst}.\ I_{norm})$

These conditions illustrate the generic condition that an extremely healthy individual (not neurotic at all) that encounters only events that are less intense than its immunity level will never develop a relapse.

Analysis of Case $e > I$ and $\beta_{neu}\, neu = lSt$: In this case the equation becomes:

$$\beta_{lst}\,\beta_{neu}\, neu = \beta_{sst}.e+(1-\beta_{sst}).neu.(1-e).(1-(w_1.(astv_{norm} - (1-\alpha_{astv}).\, neu. \\ astv_{norm})(1-neu)) +w_2(astv_{norm} - (1-\alpha_{astv}).\, neu.\, astv_{norm}) \\ (1-\alpha_{avc}.neu))+w3.(astv_{norm} - (1-\alpha_{astv}).\, neu.\, astv_{norm})))$$

Rewriting this equation in general, provides an equation of third degree, which for given values of the parameters can be solved in an algebraic manner or numerically. For some special cases of parameter values the equation becomes simpler. For example, when $\alpha_{astv} = 1$, it becomes a quadratic equation:

$$\beta_{lst}\,\beta_{neu}\, neu = \beta_{sst}.e+(1-\beta_{sst}).neu.(1-e).(1-(w_1.\ astv_{norm}\,(1-neu) + \\ w_2\ astv_{norm}\ (1-\alpha_{avc}.neu))+w3.\ astv_{norm}\ neu.\ astv_{norm})))$$

This situation describes how an individual that encounters events which are more intense than its immunity level will not change, if his long-term stress level is in balance with his level of neurotiscm.

6 Conclusion

The grand challenge addressed in the research that is reported in this paper is to develop a software agent that is capable of monitoring individuals' condition in certain events. In this paper a first step has been taken. A model has been developed that is able to explain the onset of recurrence and relapse based on personal characteristics and stressor events. The proposed model is heavily inspired by scientific findings about the relapse or recurrence onset. Having this foundation, a formal model has been developed and used to simulate different individuals' situations, which corresponded to their personality and characteristics. A mathematical analysis has been performed to demonstrate the occurrence of equilibrium conditions, fundamentally beneficial to describe convergence and stable state of the model. The proposed model provides a basic building block in designing a software agent that will support the human. Future work of this agent and model integration will be specifically focus how interactions and sensing properties can be further developed and enriched, to promote a better way to fluidly embedded this into any monitoring and health informatics system.

References

1. Belsher, G., Costello, C.G.: Relapse after Recovery from Unipolar Depression: A Critical Review. Psychological Bulletin 104 (1988)
2. Both, F., Hoogendoorn, M., Klein, M.A., Treur, J.: Formalizing Dynamics of Mood and Depression. In: Ghallab, M., Spyropoulos, C.D., Fakotakis, N., Avouris, N. (eds.) Proc. of the 18th European Conf. on Artificial Intelligence, ECAI 2008, pp. 266–270. IOS Press, Amsterdam (2008)
3. Brilman, E.I., Ormel, J.: Life Events, Difficulties, and Onset of Depressive Episodes in Later Life. Psychological Medicine 31 (2001)
4. Cohen, S., Wills, T.A.: Stress, Social Support, and the Buffering Hypothesis. Psychological Bulletin 98, 310–357 (1985)
5. Gunthert, K.C., Cohen, L.H., Armeli, S.: The Role of Neuroticism in Daily Stress and Coping. Journal of Personality and Social Psychology 77, 1087–1100 (1999)
6. Kessler, R.C.: The Effects of Stressful Life Events on Depression. Annual Review of Psychology 48, 191–214 (1997)
7. Picard, W.R., Vyzas, E., Healey, J.: Towards Machine Emotional Intelligence: Analysis of Affective Physiological State. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(10) (2001)
8. Shahar, G., Joiner, J., Zuroff, D.C., Blatt, S.J.: Personality, Interpersonal Behavior, and Depression: Co-Existence of Stress-Specific Moderating and Mediating Effects. Personality and Individual Differences (36), 1583–1596 (2004)
9. Ormel, J., Wohlfarth, T.: How Neuroticism, Long Term Difficulties, and Life Situation Change Influence Psychological Distress: A Longitudinal Model. Journal of Personality and Social Psychology 60(5), 744–755 (1991)
10. Monroe, S.M., Harkness, K.L.: Life stress, the kindling hypothesis and the recurrence of depression: consideration from a life stress perspective. Psy. Review 112(2), 417–445 (2005)
11. Lewinsohn, P.M., Hoberman, H., Teri, L., Hautzinger, M.: An integrative theory of depression. In: Reiss, S., Bootzin, R.R. (eds.) Theoretical Issues in Behavior Therapy, pp. 331–352 (1985)
12. Ingram, R.E., Luxton, D.D.: Vulnerability-Stress Models. In: Abel, J. (ed.) Development of Psychopathology: Stress-Vulnerability Perspectives, New York (2005)
13. Bickmore, T., Gruber, A., Picard, R.: Establishing the Computer-Patient Working Alliance in Automated Health Behavior Change Interventions. Patient Education and Couseling 59(1), 21–30 (2005)
14. Kessing, L.V., Hansen, M.G., Andersen, P.K., Angst, J.: The predictive effect of episodes on the risk of recurrence in depressive and bipolar disorders - a life-long perspective. Acta Psychiatrica Scandinavica 109, 339–344 (2004)

Plan Repair in Conflict-Free Routing

Adriaan ter Mors[1] and Cees Witteveen[2]

[1] Almende BV, Rotterdam, The Netherlands
adriaan@almende.org
[2] Delft University of Technology, The Netherlands
c.witteveen@tudelft.nl

Abstract. In conflict-free routing a set of agents have to traverse a common infrastructure without interfering with each other. Maza and Castagna [1] showed how the route plans of such agents can be repaired by maintaining the priority of agents on infrastructure resources. They also developed an algorithm that allows agents to change priorities to avoid long waits. We extend the work of Maza and Castagna by (*i*) specifying an algorithm that allows more priority changes, and by (*ii*) defining a graph structure that can predict exactly which priority changes will lead to a deadlock, and which will not.

1 Introduction

Let $\mathcal{A}$ be a set of vehicles or *agents* each with a start location and a goal location. The problem of conflict-free routing is to find the shortest-time route to the goal location for each of the agents, in such a way that agents do not interfere with each other (e.g., they do no collide). This problem has applications in the routing of Automated Guided Vehicles (AGVs) in warehouses or at container terminals in harbors, but also in the routing of aircraft on taxiways.

In the literature many *planning approaches* to this problem exist, that find an optimal (shortest-time) plan for a *single* agent, given the plans of previous agents, which may not be invalidated. Early algorithms include those from Fujii et al. [2], and from Kim and Tanchoco [3], who proved that their algorithm returned the optimal solution in $O(|\mathcal{A}|^4|R|^2)$ time, where $\mathcal{A}$ is the set of agents, and R is the set of infrastructure resources. More recent work includes that of Hatzack and Nebel [4], who presented a sub-optimal $O(|\mathcal{A}||R|)$ algorithm, and an algorithm from Ter Mors et al. [5] that finds an optimal solution in $O(|\mathcal{A}||R|\log(|\mathcal{A}||R|) + |\mathcal{A}||R|^2)$ time.

In of all of the aforementioned application domains, deadlock situations can occur if only a single agent is delayed. One approach is to detect and resolve deadlocks (cf. [6]) as they occur, but that leaves the question what to do with the plans that have been made: it is unclear whether these are still of good quality, or whether they will soon lead to another deadlock. Many other approaches are based on *deadlock avoidance*, with little or no planning involved. In these approaches, agents usually take the shortest *path*, and before an agent enters the next road or intersection, a check is performed to see if it safe to do so. Wu and

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 46–55, 2009.

Zhou [7], for example, build a Petri net model of the transportation system, and only allow an agent to enter the next segment if it leaves the Petri net in a live state. Reveliotis [8] presents a similar approach in which he extends Dijkstra's Banker's algorithm. The disadvantage of these deadlock avoidance approaches is that they do not try to find optimal agent routes.

The approach we discuss in this paper was first explored by Maza and Castagna [1]. They assume a planning algorithm such as [3], and they note that the set of agent plans specifies for each infrastructure resource the order in which the agents should enter the resource. They prove that if during execution we simply maintain this *priority* order, then no deadlocks can occur. A disadvantage is that it requires agents to wait for delayed agents. The authors came up with an algorithm that sometimes allows non-delayed agents to increase their priority. However, the constraints under which this is possible are quite restrictive.

We extend the approach of Maza and Castagna in two ways (in section 4): first, we define an algorithm that can predict exactly which priority changes will lead to a deadlock, and which will not; second, we present an algorithm that allows more priority changes. In section 2 we will first present a model for conflict-free routing that is based on *resources* with a finite capacity. Next, in section 3, we will discuss how agents can execute their route plans in a safe and deadlock-free manner. Finally, in section 5, we present experiments that compare our algorithm to those of Maza and Castagna.

2 Model

An *infrastructure* is an undirected graph $G = (V, E)$, where V is a set of vertices representing *intersections*, and $E \subseteq V \times V$ is a set of edges representing *lanes*. For each agent $A_i \in \mathcal{A}$ there is a pair (s_i, d_i) of locations were s_i is the agent's start location and d_i its destination location. We will treat both lanes and intersections as *resources* that must be traversed by the agents, in non-zero time. Hence, we define the set R of resources by $R = V \cup E$. The function $\mathrm{C} : R \rightarrow \mathbb{N}^+$ associates with every resource $r_i \in R$ a capacity $\mathrm{C}(r_i)$ that specifies the maximum number of agents that can simultaneously occupy a resource. For an intersection resource r_i we always have $\mathrm{C}(r_i) = 1$. We also define a function $\mathrm{D} : R \rightarrow \mathbb{N}^+$ that gives the minimum travel time of a resource. From the infrastructure graph G, we derive a resource graph $R_G = (R, E_R)$ where for each edge $e = \{v_1, v_2\} \in E$, the *resource successor* relation E_R contains the pairs (v_1, e), (e, v_2), (v_2, e), and (e, v_1).

Definition 1 (Agent Plan). *Given a source, destination pair (s, d) and a resource graph R_G, an* agent plan *is a sequence $\pi = (\langle r_1, \tau_1 \rangle, \ldots, \langle r_n, \tau_n \rangle)$ of n $\langle$ resource, interval $\rangle$ pairs (called* plan steps*) such that $r_1 = s$ and $r_n = d$ and $\forall j \in \{1, \ldots, n-1\}$:*

1. *interval τ_j* meets *interval τ_{j+1},*
2. *$|\tau_j| \geq D(r_j)$,*
3. *$(r_j, r_{j+1}) \in E_R$*

The first constraint in the above definition states that the exit time of the j^{th} resource in the plan must be equal to the entry time into resource $j+1$. The second constraint requires that the agent's occupation time of a resource is at least sufficient to traverse the resource in the minimum travel time. The third constraint states that if two resources follow each other in the agent's plan, then they must be adjacent in the infrastructure.

In conflict-free routing, agents need to find plans that do not interfere with each other. Only those combinations of agent plans that do not violate any resource capacity constraints should be considered.

Definition 2 (Resource Load). *Given a set Π of agent plans, the* resource load λ *is a function $\lambda : R \times \mathbb{N} \rightarrow \mathbb{N}$ that gives the number of agents occupying a resource r_i at each time point t: $\lambda(r_i, t) = |\{\langle r_i, \tau \rangle \in \pi \mid \pi \in \Pi \wedge t \in \tau\}|$.*

In this paper we consider two additional constraints. The first is that agents are not allowed to overtake each other on a (lane) resource. The second constraint is that a resource can be used in only a single direction at one time. This implies that once an agent has entered a (lane) resource from one end (i.e., an intersection), no other agent can enter the resource from the other end until the first agent has exited. In other words, if two agents occupy the same resource at the same time, they must have entered from the same intersection. Given plan step σ, we write $ep(\sigma)$ to denote the entry point of σ, which equals the previous resource in the plan.

3 Plan Execution

We employ a simple model for agent driving: each agent has a maximum speed, and it can drive at any speed up to that maximum speed. We assume no acceleration or deceleration are necessary. Given its plan of pairs of resources and time intervals, an agent will try to traverse each resource in the specified time interval. If an agent arrives at a resource early, it will simply wait before entering the resource; if an agent is late, it will simply continue to drive along the planned sequence of resources, and it will try to make up some time. Making up time might be possible if the planned speed of traversing a resource is less than the agent's maximum speed.

With the above model of plan execution, the delay of one agent can cause a deadlock situation, as the following example illustrates.

Example 1. Consider a small airport with aircraft agents A_1 and A_2, in figure 1. A_1 starts at hangar h_2 and wants to go to h_4, while A_2 wants to go from hangar h_3 to hangar h_1. The agents make the following plans, to the effect that A_1 is allowed to pass along the taxiway straight $t_2 - t_4 - t_5$ first:

$$\pi_{A_1} = \langle h_2, 0 \rangle, \langle t_3, 2 \rangle, \langle t_2, 4 \rangle, \langle t_4, 5 \rangle, \langle t_5, 10 \rangle, \langle t_7, 11 \rangle, \langle h_4, 13 \rangle$$

$$\pi_{A_2} = \langle h_3, 0 \rangle, \langle t_6, 2 \rangle, \langle t_5, 11 \rangle, \langle t_4, 12 \rangle, \langle t_2, 17 \rangle, \langle t_1, 18 \rangle, \langle h_1, 20 \rangle$$

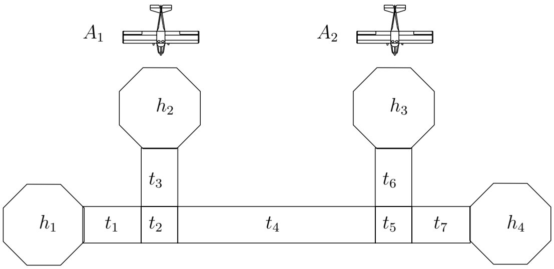


Fig. 1. Two aircraft agents A_1 and A_2 with hangars h_4 and h_1 as respective destinations

Hence, A_2 plans to stay on taxiway t_6 in the interval $[2, 11)$, just long enough to let A_1 pass. Suppose that during plan execution, A_1 stalls his engine, and he departs with a delay of 5. Then, he will be on taxiway t_4 in the interval $[10, 15)$. Clearly, this interferes with the plan of A_2, who will be on t_4 from 12 until 17. Agent A_1 and A_2 will therefore meet each other on taxiway t_4, and unless either of them can drive backwards, they will end up in a deadlock situation.

What happened in example 1 was that the *priority* that the agents had agreed upon during planning was violated during plan execution. After the planning phase, we know for each resource in which order it will be visited by the agents. If we maintain this order during plan execution, then no deadlocks can occur [9]. This leads us to formulate the following conditions for entry into a resource:

Definition 3 (Resource Entry Permission). *An agent A_i is allowed to start its next plan step $\sigma_i = \langle r_k, \tau = [t_s, t_e)\rangle$ by entering resource r_k at time t if:*

1. *σ_i is plan step number n on r_k, and $n - 1$ plan steps have been started on this resource so far,*
2. *the number of agents on r_k at time t is at least one less than $\mathrm{C}(r_k)$,*
3. *if an agent A_j occupies r_k at time t, with plan step σ_j , then $ep(\sigma_i) = ep(\sigma_j)$,*
4. *$t \geq t_s$.*

Changing the priority (i.e., the plan step number on a resource) of the agents' plan steps during plan execution can often be beneficial, for instance if A_1 in example 1 suffers a very long delay, then A_2 should go first. The question is whether we can change the priority of the agents without introducing deadlocks. One simple mechanism is from Maza and Castagna [1]: *Algorithm M2:* Grant an agent the highest priority for all the remaining resources in its plan, until the resource where the agent already has the highest priority. These priority changes will not result in a deadlock if and only if all of these resources are currently empty.

The requirements for a priority change in algorithm M2 are quite strong, which limits its applicability. In the next section, we present an algorithm that allows more priority changes.

4 Planstep-Priority Graph

We will now show how to compute which priority changes will lead to a deadlock using a *Planstep-Priority Graph (PPG).* We will construct the PPG from the plan steps of the plans of the agents. We will show that a deadlock will occur if and only if this graph contains a cycle. In the following we will write $p(r_k, \sigma_{i,x})$ to denote the priority of plan step $\sigma_{i,x}$ on resource r_k. Since priorities are based on the order in which agents visit resources, a priority of 1 is higher than a priority of 10.

Definition 4 (Enabling Plan Step). *Plan step $\sigma_{i,x+1} \in \pi_i$ is the* enabling plan step *of $\sigma_{j,y} \in \pi_j$ if $\sigma_{i,x}$ (i.e., the previous step in π_i) and $\sigma_{j,y}$ are on the same resource r_k, and:*

1. $p(r_k, \sigma_{j,y}) = p(r_k, \sigma_{i,x}) + 1 \;\wedge\; ep(\sigma_{i,x}) \neq ep(\sigma_{j,y})$*; or*
2. $p(r_k, \sigma_{j,y}) = p(r_k, \sigma_{i,x}) + \mathrm{C}(r_k) \wedge$
   $\forall h, w : p(r_k, \sigma_{i,x}) \leq p(r_k, \sigma_{h,w}) \leq p(r_k, \sigma_{j,y}) : ep(\sigma_{i,x}) = ep(\sigma_{h,w})$.

When the enabling plan step starts, at least two entry conditions from definition 3 become satisfied: first of all, there are now fewer agents occupying the resource than the capacity, and second, all agents traversing the resource in the opposite direction have exited.

Definition 5 (Planstep-Priority Graph). *Given a set of agent plans $\Pi = \{\pi_1, \ldots, \pi_n\}$, $\pi_i = (\sigma_{i,1}, \ldots, \sigma_{i,m})$, the* Planstep-Priority Graph *(PPG) is a directed graph $G = (V, E)$, where the set of vertices is given by:*

$$V = \bigcup_{i=1}^{|\mathcal{A}|} \bigcup_{j=1}^{|\pi_i|} \sigma_{i,j} \tag{1}$$

That is, there is a vertex for every plan step in every plan. The set of edges is made up of three parts $E = E_1 \cup E_2 \cup E_3$:

$$E_1 = \{(\sigma_{i,j}, \sigma_{i,j+1})\} \tag{2}$$

$$E_2 = \{(\sigma_{i,x}, \sigma_{j,y}), (\sigma_{i,x+1}, \sigma_{j,y+1}) \,|\, p(r_k, \sigma_{j,y}) = p(r_k, \sigma_{i,x}) + 1\} \tag{3}$$

$$E_3 = \{(\sigma_{i,x}, \sigma_{j,y}) |\, \sigma_{i,x} \textit{ is enabling plan step of } \sigma_{j,y}\} \tag{4}$$

The edges in the Planstep-Priority Graph reflect the order in which plan steps should be executed; if $(\sigma_{i,x}, \sigma_{j,y}) \in E$, then plan step $\sigma_{i,x}$ must *start* before $\sigma_{j,y}$ can start. Edges in E_1 express that an agent should perform the steps in its plan sequentially. The set E_2 specifies that if the priority of $\sigma_{i,x}$ (on some resource r_k) is one higher than the priority of $\sigma_{j,y}$, then $\sigma_{i,x}$ must not only start before $\sigma_{j,y}$, it must also *finish* before $\sigma_{j,y}$ (since overtaking is not allowed). This implies that the next plan step $\sigma_{i,x+1}$ of π_i must start before the next step $\sigma_{j,y+1}$ in π_j. Finally, the set E_3 states that a plan step cannot start until its enabling plan step has started.

Proposition 1. *A deadlock will occur if and only if there is a cycle in the Planstep-Priority Graph.*

Proof (sketch)
Case I: deadlock → cycle in PPG. A deadlock occurs if no agent can make any progress any more. In our model, this can occur if an agent is waiting to enter a resource, or if an agent is 'in the middle' of a resource, but stuck behind another agent. In both of these cases, we can find a path of edges in E to connect a waiting agent to the culprit. Because a deadlock involves a cyclic wait, the paths in E between waiting agents form a cycle.

Waiting to enter a resource: Definition 3 gives three reasons why an agent should wait for entry; two of them indicate that an agent's enabling plan step has not started yet (in which case we can find an edge in E_3). The third reason is that it's not yet the agent's turn to enter a resource; in this case, we can construct a path between the waiting agents using edges from E_1 and E_2.

Stuck behind another agent: Using edges in E_2, we can find a path to the first agent to exit the resource, and this agent is waiting to enter its next resource, so we return to the previous case.

Case II: cycle in PPG → deadlock. The edges in PPG specify a precedence order between plan steps; if this order is cyclic, then no plan step in a cycle can ever start, and hence a deadlock will occur the moment an agent tries to start a plan step that is part of a cycle.

4.1 Algorithm: Increasing Agent Priorities

Arriving at a resource, an agent A_i may find that there are one or more agents with a higher priority that have not entered the resource yet. To continue driving, agent A_i should increase its priority to the maximum of all these delayed agents. If we only change priorities on the current resource, however, then it is easy to create a deadlock situation. Suppose that one of the delayed agents, agent A_j, follows the same route as (what remains of the plan of) A_i, but in the opposite direction. If agent A_i is allowed to increase its priority, then sooner or later agent A_i and agent A_j will meet, when neither can make progress anymore. Hence, we should check for each of the delayed agents whether they also have a higher priority on subsequent resources in A_i's plan, and, if yes, then A_i can only increase its priority if these agents have not reached these resources yet.

From line 8 of algorithm 1, we can see that the set A of delayed agents can change as A_i scans the resources that remain in its plan. A new agent A_j can be added to A if for some future resource r_k the priority of A_j is higher than that of A_i, but lower than that of some agent already in A. Of course, an agent can also disappear from the set A as soon as it does not share any resources anymore with agent A_i. In most cases, the procedure described above results in a deadlock-free change of priorities. In a few cases, however (most notably when a delayed agent's path crosses that of A_i more than once), it is still possible that a deadlock situation is created. Therefore, we simply check after increasing an agent's priority whether the Planstep-Priority Graph is still acyclic.

Algorithm 1. Increase Agent Priority

```
Require: agent $A_i$, plan $\pi_i$ and next plan step $\sigma_{i,k}$, Planstep-Priority Graph PPG
Ensure: give agent $A_i$ the highest priority on the next resource in its plan $\pi_i$, in case
    this does not create a deadlock
$n \leftarrow \text{entryCounter}(r_{i,k})$
$A \leftarrow \{A_j \in \mathcal{A} \mid n < p(r_{i,k}, \sigma_{j,x}) < p(r_{i,k}, \sigma_{i,k})\}$
$M \leftarrow \varnothing$
deadlock ← false
while $A \neq \varnothing \wedge \neg$deadlock do
    $r \leftarrow r_{i,k}$
    $A_m \leftarrow \min_{(A_j \in A)} p(r, \sigma_{j,x})$
    $A \leftarrow \{A_j \in \mathcal{A} \mid p(r, \sigma_{m,x}) \leq p(r, \sigma_{j,y}) < p(r, \sigma_{i,k})\}$
    for all $A_j \in A$ do
        if locatedAt$(A_j, r)$ then
            deadlock ← true
            continue
    $M \leftarrow M \cup \langle r, A \rangle$
    $k \leftarrow k + 1$
if $\neg$deadlock then
    for all $\langle r, A \rangle \in M$ do
        increasePriority$(A_i, A, r, \text{PPG})$
    if cycle in PPG then
        rollbackPriorityChanges$(M, \text{PPG})$
```

5 Experiments

In this section, we will evaluate the performance of our IAP algorithm in comparison with algorithms 1 and 2 from Maza and Castagna (we will refer to these algorithms as M1 — which is to perform no priority changes — and M2 respectively). The first question we aim to answer is: *which algorithm allows the most priority changes?* The second question is: *which algorithm results in plan execution with the smallest delay?* Intuitively, we would expect that the algorithm that performs the most changes will result in the smallest delay, because with every change an agent with little or no delay is given priority over an agent with more delay.

To introduce delay into the system, we introduce random incidents that temporarily immobilize an agent. The *incident rate* is a value between 0 and 1 that indicates the probability that an agent will suffer an incident during the execution of one plan step. Hence, if the incident rate equals 0.1, and an agent's plan contains 50 plan steps, then the agent can expect to have five incidents on the way. The *incident duration* is always the same for one experiment run.

5.1 Setup

We experimented with a set of 100 random infrastructures, each consisting of 120 intersections and 200 lanes. To evaluate which algorithm results in the smallest delay, we measure the (relative) *mechanism delay*: the time an agent has to wait for one or more conditions of definition 3 to become satisfied (or if an agent is waiting behind another agent that is accumulating mechanism delay), as a percentage of its plan length.

In our experiments we varied three parameters: the number of agents, from 100 to 500, with an increment of 10; the incident rate, with values 0.02, 0.06, and 0.1; and the incident duration, with values of 30 and 60 seconds. A problem instance for a single experiment run consists of: (*i*) one of the 100 infrastructures, chosen by: experiment-number modulo 100; (*ii*) for each agent a randomly chosen start and destination location; (*iii*) a set of agent plans, obtained by letting each agent plan its optimal route, given the reservations of the agents that have planned before it; (*iv*) for each step of each agent plan, we use the incident rate parameter to determine whether the agent will suffer an incident during the execution of this plan step.

5.2 Results

In figure 2, we compare the number of priority changes performed by algorithms IAP and M2, for three different parameter settings: an incident rate of 0.02, an incident rate of 0.06, and an incident rate of 0.1; the incident duration was set to 60 seconds. The conclusions that can be drawn from figure 2 are straightforward: (*i*) IAP performs more changes than M2, sometimes up to twice as many; (*ii*) the number of changes performed by both of the algorithms increases linearly with the number of agents.

Figure 3 shows the relative mechanism delay (i.e., the mechanism delay divided by the plan length), averaged over the six different incident settings. We can see from figure 3 that IAP results in lower mechanism delay than algorithms M1 and M2. However, the differences are perhaps not as large as we may have anticipated after seeing figure 2: IAP often performs twice as many priority changes as M2, but the difference in performance ranges from 0% to 10% of relative mechanism delay. Moreover, algorithm M1, which does not perform any priority changes at all, is actually quite competitive for larger numbers of agents in the system.

Table 1 shows the relative mechanism delay for each of the six incident settings. For each combination of incident setting and number of agents, the algorithm that produced the lowest delay is highlighted in bold. A quick glance at the table reveals that for settings with a low incident load, IAP (and to a lesser extent, M2) performs very well. For the highest incident load (rate = 0.1 and duration = 60s), however, M2 and M1 perform better, with the latter best of all for more than 300 agents. From this table we can conclude that for higher incident loads and more congestion, there is less benefit from making priority changes. A reason might be that for high incident rate scenarios, there is no longer a clear distinction between agents that are on time and agents that are delayed. Moreover, there is a good chance that an agent will suffer an incident shortly after it has obtained a priority increase. In that case, the priority change might do more harm than good.

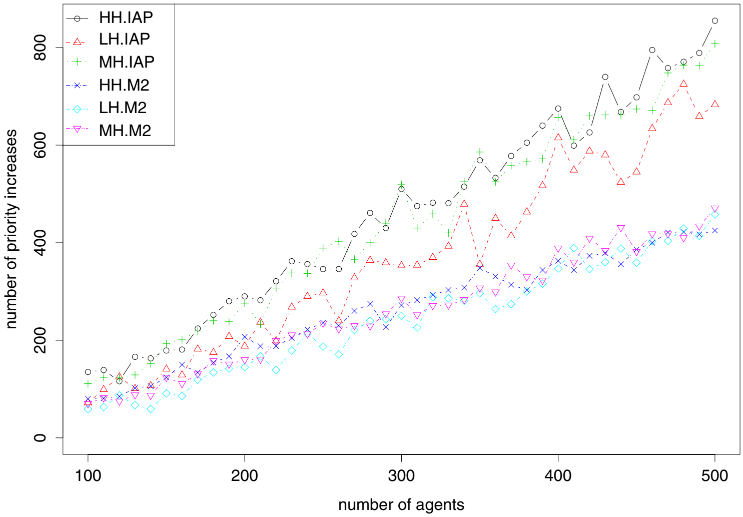


Fig. 2. Number of priority changes made by IAP and algorithm M2 for incident settings: HH = (0.1, 60s) ; MH = (0.06, 60s) ; LH = (0.02, 60s)

Table 1. Relative mechanism delay for six different incident settings

$\|\mathcal{A}\|$	0.02, 30s			0.02, 60s			0.06, 30s			0.06, 60s			0.1, 30s			0.1, 60s		
	M1	M2	IAP	M1	M2	IAP	M1	M2	IAP	M1	M2	IAP	M1	M2	IAP	M1	M2	IAP
100	14	3	**2**	36	8	**4**	30	**20**	25	53	35	**28**	28	16	**9**	44	**25**	27
150	15	10	**7**	33	21	**10**	27	10	**6**	36	**22**	26	24	13	**10**	40	**31**	32
200	14	11	**5**	38	11	**8**	17	12	**9**	44	35	**30**	22	26	**14**	34	**25**	26
250	16	10	**6**	30	27	**18**	17	11	**6**	36	32	**24**	18	13	**5**	33	28	**28**
300	11	5	**3**	30	18	**5**	19	13	**13**	37	36	**36**	21	**15**	15	**32**	35	45
350	11	6	**4**	30	16	**9**	16	18	**13**	31	31	**30**	17	**14**	14	**27**	31	28
400	9	4	**3**	32	24	**16**	15	11	**6**	29	26	**21**	**16**	27	20	**25**	29	35
450	10	5	**3**	26	16	**11**	16	11	**8**	26	27	**22**	18	13	**11**	**25**	39	43
500	12	5	**3**	28	19	**10**	14	15	**8**	25	**20**	21	17	14	**8**	**22**	24	33

5.3 Discussion

The rationale behind making priority changes in the manner of algorithms M2 and IAP is that a 'timely' agent can continue execution of its plan unhindered, whereas the delayed agent might not even notice the loss of priority. Of course, this is only a rule of thumb, and, as we have seen in figure 3 and table 1, it does not always hold, and sometimes priority changes can have averse effects.

The added value of IAP over M2 is therefore better illustrated by figure 2 than by figure 3. In figure 2 we see that IAP finds more possibilities for changing priorities than M2. In future work, we will come up with additional heuristics to determine which priority changes are beneficial and which are detrimental. On the basis of such information we can extend IAP to an efficient and effective plan repair algorithm.

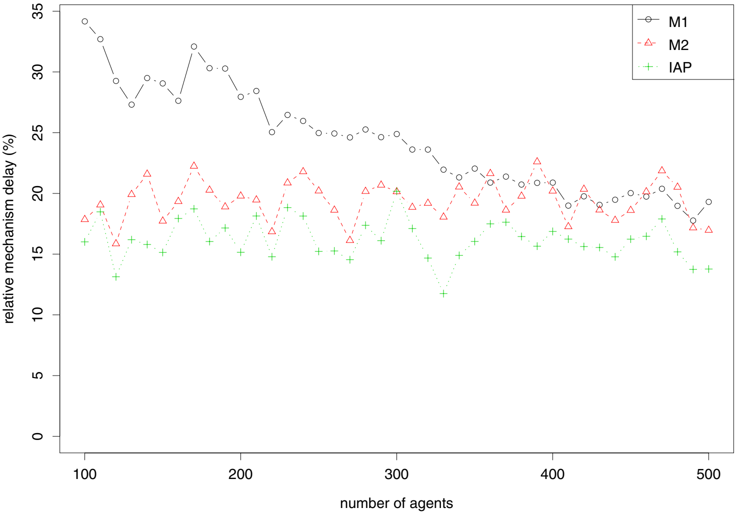


Fig. 3. Relative mechanism delay, averaged over different incident settings

References

1. Maza, S., Castagna, P.: A performance-based structural policy for conflict-free routing of bi-directional automated guided vehicles. Computers in Industry 56(7), 719–733 (2005)
2. Fujii, S., Sandoh, H., Hozaki, R.: A routing control method of automated guided vehicles by the shortest path with time-windows. In: Production Research: Approaching the 21st Century, pp. 489–495 (1989)
3. Kim, C.W., Tanchoco, J.: Conflict-free shortest-time bidirectional AGV routeing. International Journal of Production Research 29(1), 2377–2391 (1991)
4. Hatzack, W., Nebel, B.: The operational traffic problem: Computational complexity and solutions. In: Cesta, A. (ed.) Proceedings of the 6th European Conference on Planning (ECP 2001), pp. 49–60 (2001)
5. ter Mors, A.W., Zutt, J., Witteveen, C.: Context-aware logistic routing and scheduling. In: Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling, pp. 328–335 (2007)
6. Lehmann, M., Grunow, M., Günther, H.O.: Deadlock handling for real-time control of AGVs at automated container terminals. OR Spectrum 28(4), 631–657 (2006)
7. Wu, N., Zhou, M.: Resource-oriented petri nets in deadlock avoidance of AGV systems. In: Proceedings of the 2001 IEEE International Conference on Robotics and Automation (ICRA), Seoul, Korea, May 2001, pp. 64–69 (2001)
8. Reveliotis, S.A.: Conflict resolution in AGV systems. IIE Transactions 32(7), 647–659 (2000)
9. Maza, S., Castagna, P.: Conflict-free AGV routing in bi-directional network. In: Proceedings of the 8th IEEE International Conference on Emerging Technologies and Factory Automation, Antibes-Juan les Pins, France, October 2001, vol. 2, pp. 761–764 (2001)

Intelligent Decoupled SAC-SVD Method in Color Space Transformation of Computer Vision

Jian-Long Kuo

Institute of M.A.E. System Information and Control, National Kaohsiung First University of Sci. and Tech., Nan-Tze, Kaoh-Siung 811, Taiwan
jlkuo@ccms.nkfust.edu.tw

Abstract. This paper proposes a decoupled SAC-SVD method to calculate color space transformation in single assignment C (SAC) language. Chromaticity coordinate transformation is discussed based on the decoupled SAC-SVD method. The conventional color space transformation is decomposed into three independent equations, respectively. Singular value decomposition (SVD) transformation is proposed to provide the constant values for the associated derivation. Such an algorithm is suitable for the arithmetic operation of single assignment C. The hardware description language VHDL can be generated by using Data Flow Graph in a single assignment C compiler. The results will show that the proposed algorithm can shorten the execution time to a certain extent. It is evident that the proposed intelligent algorithm can be very helpful in implementing the hardware-interface or software-driver for the computer vision.

Keywords: Decoupled, reconfigurable computing system (RCS), single assignment C (SAC), parallel computing, VHDL, color space, chromaticity coordinate, data flow graph (DFG), singular value decomposition (SVD).

1 Introduction

Recently, systems on a chip, such as FPGA/CPLD, have drawn considerable attention in consumer electronics IC design as they represent a powerful IC design and development platform. However, the design language, such as VHDL or Verilog, still has a tedious implementation procedure for a complicated high-level signal-processing algorithm when the embedded system is considered.

It is not easy to realize a high-level signal-processing algorithm directly by using VHDL or Verilog and a powerful language is required to realize complicated signal processing algorithms or embedded systems. A variety of C language becomes more and more popular due to easy programming via SystemC or HandleC.

Recently, the SA-C language was developed due to the VHDL design requirement [1]-[5]. Research on the Cameron Project, originally running at Colorado State University, has rapidly spread worldwide. The purpose of the Cameron project is to make FPGAs and other adaptive computing systems available to more applications' designers.

To this end, SA-C has been developed into an alternative of C programming language and an optimizing compiler for the parallel operation [6]. The language and compiler can transfer high-level programs directly into FPGA design, especially on

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 56–65, 2009.

complicated signal processing or image processing, as well as many other system integration applications. As shown in Figs. 1 and 2, the SA-C source programs were initially translated into DFG, which is a token-driven semantics [7]. SA-C can be further translated into a type of graph called the data dependence and control flow (DDCF) graph, which is convenient for advanced optimization.

A conventional DFG, however, does not have nodes with internal state, for example, registers. Therefore, yet another round of optimization and translation transforms the data flow graph into an abstract hardware architecture graph, which is a data flow graph with state full nodes (mostly registers) and hand-shaking signals. The abstract hardware graph is then optimized one last time before VHDL is generated.

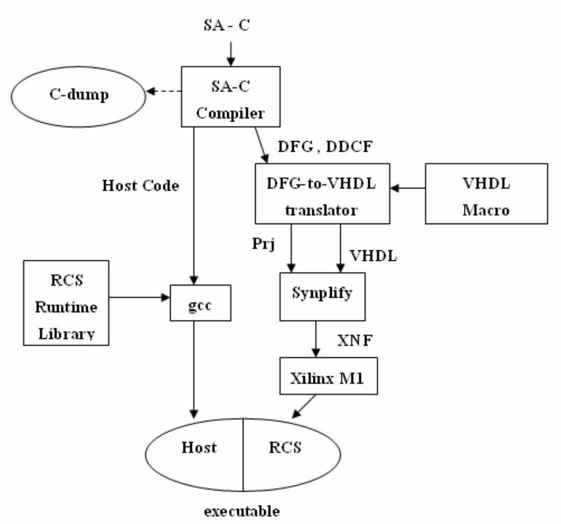


Fig. 1. SAC compiler procedure for the reconfigurable computing system

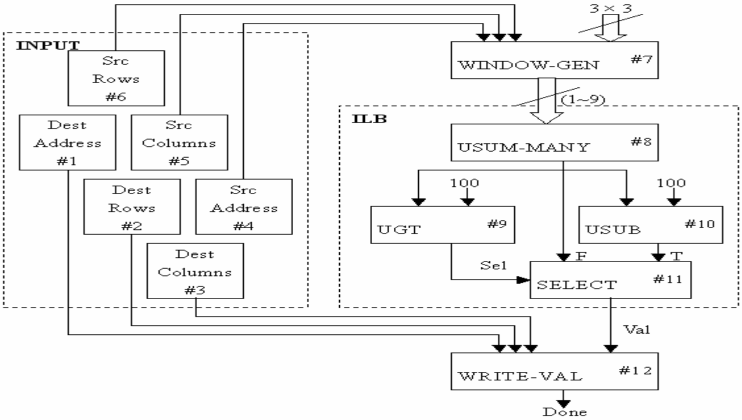


Fig. 2. Paradigm for the DFG generation of the **for** loop

2 Representation of Arrays in SA-C

In general, the arrays in SA-C can be represented by two vectors. A data vector contains all array elements in row-major order, and a shape vector specifies the number of elements per axis. Let **A** be an n-dimensional array with shape vector and data vector respectively.

$$\mathbf{sv} = [sv_0, sv_1, \ldots, sv_{n-1}], \mathbf{dv} = [dv_0, dv_1, \ldots, dv_{l-1}] \cdot \tag{1}$$

Then the length l of the data vector should be

$$l = \prod_{k=0}^{n-1} sv_k \,. \tag{2}$$

Sub-arrays or elements of the array may be addressed by index vectors of the set:

$$\begin{aligned} \{[iv_0, iv_1, \ldots, iv_{m-1}] \mid (0 \le m \le n) \wedge \\ (\forall k \in \{0,1,\ldots,m-1\} : 0 \le iv_k \le sv_k)\} \end{aligned} \,. \tag{3}$$

An index vector $\mathbf{iv} = [iv_0, iv_1, \ldots, iv_{m-1}]$ selects the sub-array with shape vector $[sv_m, sv_{m+1}, \ldots, sv_{n-1}]$ and data vector: $[dv_p, dv_{p+1}, \ldots, dv_{p+q-1}]$, where p and q satisfy:

$$p = \sum_{k=0}^{m-1} (iv_j \prod_{t=k+1}^{n-1} sv_t) \quad \mathrm{q} = \prod_{k=m}^{n-1} sv_k \,. \tag{4}$$

The special cases (m=0) and (m=n) specify the selection of the whole array and the selection of the single array element dv_p respectively.

3 Matrix Operation Algorithm for SA-C

SA-C has modified data type signed or unsigned integers and fixed-point numbers with user-defined bit-widths. Nevertheless, SA-C has multi-dimensional rectangular arrays so that its extents are determined dynamically or statically. For example, the type declaration **`int14 M[:,6]`** is a declaration of a matrix M of 14-bit signed integers. The left dimension is determined dynamically, while the right dimension is specified by the user.

What is most important is the **`for`** loop structure. The **`for`** loop has three parts: a generator, a body, and a return expression. By using the **`for`** loop in SA-C, many parallel matrix operations can be realized very easily, e.g.,

```
 for _ in [n] return  (tile (B))
for _,_ in [m,n] return (tile (B))
```

will produce **`m`** and **`n`** iterations without having to declare an iteration variable. This will be helpful to the matrix expansion or flattening discussed in this paper.

Type I: Flattening a matrix to a vector:

```
int16[:] main (int16 A[:,:]) return(
for V (~,:) in A return (tile (V))  );
```

Type II: Restructuring a vector to a matrix:

```
int16[:,:] main (int16 A[:], int16 n)
{int16 s=extents(A);
assert (s%n==0, "not rectangular",s,n);
int16 res[:,:]=for window V[n] in A step 5:
(n)return (array(V)); }  return(res);
```

4 Chromaticity Coordinate Transformation from sRGB to YUV Transformation

The YUV model defines a color space in terms of one luminance and two chromaticity components. YUV is used in the PAL and NTSC systems of video signal for the processor system and it is the standard in much of the world. The YUV models human perception of color more closely than does the standard RGB model used in processor graphics hardware, but not as closely as HSL color space and HSV color space. Y stands for the luminance component and U and V are the chromaticity components. The YCbCr or YPbPr color space used in video interface, are derived from YUV. Cb/Pb and Cr/Pr are simply scaled versions of U and V, and are sometimes inaccurately called YUV.

YUV signals are created from an original RGB source. The weighted values of R, G and B are added together to produce a single Y signal, representing the overall brightness, or luminance, of the pixel. The U signal is then created by subtracting Y from the blue signal of the original RGB, and then scaling. Also, V is created by subtracting Y from the red color, and then scaling by a another factor. The following equations can be used to derive *Y*, *U* and *V* from *R*, *G* and *B* in a compact matrix form:

$$\mathbf{C}_{\mathbf{YUV}} = \mathbf{M}_{\mathbf{YR}}\mathbf{C}_{\mathbf{sRGB}} \ . \tag{5}$$

In an expansion form, the above matrix form can be expressed as:

$$X = m_{11}R + m_{12}G + m_{13}B, \quad Y = m_{21}R + m_{22}G + m_{23}B$$
$$Z = m_{31}R + m_{32}G + m_{33}B$$

The standard elements of the above matrix can be described in detail as follows:

$$\begin{aligned} &Y=(0.299)\,R+(0.587)G+(0.114)B \\ &U=0.492(B-Y)=(-0.147)\,R-(0.289)G+(0.436)B. \\ &V=0.877(R-Y)=(0.615)\,R-(0.515)G-(0.100)B \end{aligned} \tag{6}$$

$\mathbf{M}_{\mathbf{YR}}$ is defined with respect to the different color temperature conditions.

5 Singular Value Decomposition (SVD) Method

The Singular value decomposition (SVD) method has been used as the solution for the nonlinear systems such as image processing in the optical processing system. The advantage of SVD is that this method does not require the inverse matrix operation.

Nevertheless, the SVD can still keep the matrix transformation valid. The transformation can work very well.

If the elements of the modal matrix are defined as the fraction form, the denominator is possibly equal to zero under some specific rotating angle. This might induce the irregular problem. The inverse of matrix $\mathbf{K}_t$ does not exist in some specific angles. The above similarity transformation cannot be used. Further generalized formulation has to be derived by using singular value decomposition. Therefore, SVD is used to deal with this problem.

Assume that the matrix $\mathbf{M_{YR}}$ can be the dimension of $m \times n$ with $m \geq n$, then the $\mathbf{M_{YR}}$ can be decomposed into the following form with m=n=3,

$$\mathbf{M_{YR}} = \mathbf{M}_t \mathbf{M'_{YR}} \mathbf{N}_t^H, \tag{7}$$

where $\mathbf{M}_t$ is a 3×3 matrix and $\mathbf{N}_t$ is a 3×3 matrix. $\mathbf{M}_t^H$ is a conjugate transpose matrix of $\mathbf{M}_t$ and $\mathbf{N}_t^H$ is a conjugate transpose matrix of $\mathbf{N}_t$.

To find the SVD for the $\mathbf{M_{YR}}$ with $m \times n$ dimension, the following steps are required.

a) In this paper, the rank is *rank* ($\mathbf{M_{YR}}$)=r=m=n. Therefore, the *rank* ($\mathbf{M^T{}_{YR}M_{YR}}$) =r=m=n. In this case, m=n=2.
b) Define the singular value: $\sigma_i = \sqrt{\lambda_i(\mathbf{M_{YR}}^H \mathbf{M_{YR}})}$, $i = 1 \sim n$.
c) $\lambda_i v_i = \mathbf{M_{YR}}^T \mathbf{M_{YR}} v_i$, i=1~n. Determine the right eigenvector v_i of the $\mathbf{M_{YR}}^T \mathbf{M_{YR}}$ to form the right modal matrix $\mathbf{N}_t$.
d) Determine the left eigenvector of $u_i = (1/\sigma_i)\mathbf{M_{YR}} v_i$, i=1~n, to form the left modal matrix $\mathbf{M}_t$.
e) The $\mathbf{M'_{YR}} = \text{diag}(\sigma_1, \sigma_2, ..., \sigma_n)$.

It is interesting to mention that if the $\mathbf{M}_t$ and $\mathbf{N}_t$ are chosen as unitary matrix with scaling normalization factor, both of the two matrices have orthogonal columns so that:

$$\mathbf{M}_t^H \mathbf{M}_t = \mathbf{N}_t^H \mathbf{N}_t = \mathbf{I}. \tag{8}$$

The $\mathbf{M'_{YR}}$ is a diagonal matrix again based on SVD method. In general, the two modal matrices are not necessarily always the same.

5.1 Generalized Derivation for the Color Space Transformation

To describe the above transformation in detail, the generalized equation can be written in the following form, with three factors:

$$\begin{aligned}\mathbf{M}'_{\mathbf{YR}} &= \mathbf{N}_{\mathrm{t}}^{\mathrm{H}}\mathbf{M}_{\mathbf{YR}}\mathbf{M}_{\mathrm{t}} \\ &= \begin{bmatrix} k_{11} & 0 & 0 \\ 0 & k_{22} & 0 \\ 0 & 0 & k_{33} \end{bmatrix}\end{aligned} \tag{9}$$

where k_{11}, k_{22}, and k_{33} are constant values which can be pre-determined before performing the arithmetic operation.

$$\mathbf{C}'_{\mathbf{sRGB}} = \mathbf{M}_{\mathrm{t}}\mathbf{C}_{\mathbf{sRGB}}, \mathbf{C}'_{\mathbf{YUV}} = \mathbf{M}_{\mathrm{t}}\mathbf{C}_{\mathbf{YUV}}. \tag{10}$$

$$\mathbf{C}_{\mathbf{sRGB}} = \mathbf{N}_{\mathrm{t}}^{\mathrm{H}}\mathbf{C}'_{\mathbf{sRGB}}, \mathbf{C}_{\mathbf{YUV}} = \mathbf{N}_{\mathrm{t}}^{\mathrm{H}}\mathbf{C}'_{\mathbf{YUV}} \tag{11}$$

The original color space transformation can be rewritten as:

$$\mathbf{C}'_{\mathbf{YUV}} = \mathbf{M}'_{\mathbf{YR}}\mathbf{C}'_{\mathbf{sRGB}} \tag{12}$$

where the original matrix form can be expressed as follows:

$$\begin{aligned}&X' = m_{11}'R' + m_{12}'G' + m_{13}'B', \quad Y' = m_{21}R' + m_{22}'G' + m_{23}B' \\ &Z' = m_{31}R' + m_{32}'G' + m_{33}'B'\end{aligned} \tag{13}$$

Since the $\mathbf{M}'_{\mathbf{YR}}$ is decoupled, it is very simple to express the above equations as the following three arithmetic forms:

$$X' = m_{11}'R', \ \ Y' = m_{22}'G', \ \ Z' = m_{33}'B' \tag{14}$$

Therefore, the color space transformation becomes three simple independent equations instead of a coupled matrix form. The computation speedup of image processing is achieved very easily. The computation burden of image processing can be reduced to a considerable extent.

Since three equations are in the linear combination, the factors can be specified further under the above-mentioned steps. Any software-based driver or hardware-based processor can implement the algorithm very easily by using only fixed-point and integer data type in SAC. The algorithm is based on the fixed-point operation. No matrix data type related to matrix operation is required for the proposed method.

The fixed-point processor or the FPGA/CPLD chip in the interface card can adapt the algorithm very easily by using the SAC compiler. Since the operation is decomposed into three independent equations, the calculation can be further implemented by parallel operation without any coupling terms.

6 Verifications

6.1 Comparison with Execution Speed

Color space transformations are often applied in many processor peripheral and interface cards. Therefore, the proposed algorithm requires rapid computing to fulfill the real-time image-processing requirement.

The SA-C compiler can generate the parallel computing VHDL for the circuit-level models of the associated algorithm. Some speedup tests are illustrated to show the capability of fast computing.

As shown in Table 1, the verification illustrates three cases for study: the ANSI-C VC program running on PC and SA-C running on the LINUX system for PC. Two of the realization approaches were compared: the proposed decoupled arithmetic computation and the conventional coupled matrix computation. The following cases were verified on LINUX system on the 1GHz Pentium PC. The VHDL were verified on Xilinx FPGA platform under 40MHz clock.

Case 1: Algorithm run using SA-C with fix data type; SA-C compiler compiles proposed decoupled arithmetic computation for proposed algorithm.

Case 2: Algorithm run using SA-C with float data type; SA-C compiler also compiles conventional coupled matrix computation.

Case 3: Algorithm run using ANSI-C with float data type; ANSI-C VC compiler compiles conventional coupled matrix computation.

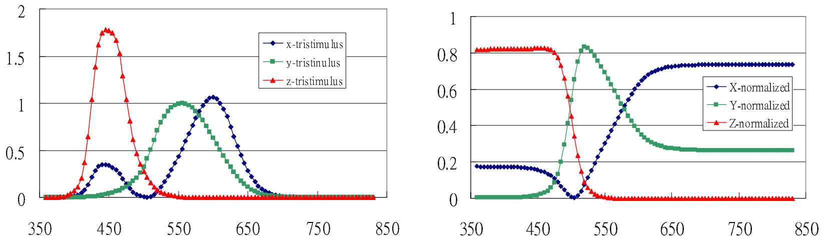
Table 1. Performance comparison for the case study

Case	Pentium PC	percentage	FPGA	percentage
Item	Computation time, ms	cycles	Computation time, ms	Flip-flops
Case 1	4	15%	0.1	15%
Case 2	30	46%	0.19	46%
Case 3	60	49%	0.48	49%

As shown in Table 1, the comparison shows that Case 1 spends less time to calculate the algorithm. Case 3 spends more time to calculate the algorithm. Results show that the proposed algorithm combined with the SA-C programming will be faster than the conventional programming using SA-C only. Conventional SA-C programming will declare the numerical operation by using float data type directly. By verification, the proposed algorithm is faster than the conventional technique. SA-C programming also proved to be faster than the conventional ANSI-C VC language that is sequentially programmed. The sequential C programming technique might have critical compatibility problems with circuit-level models by VHDL description. VHDL can execute the computing based on the logic design which is considered in the circuit level. While the sequential C programming has to execute the instruction within one instruction cycle, SA-C has the reconfigurable capability to perform the parallel computing.

6.2 Computing of Color Space Transformation

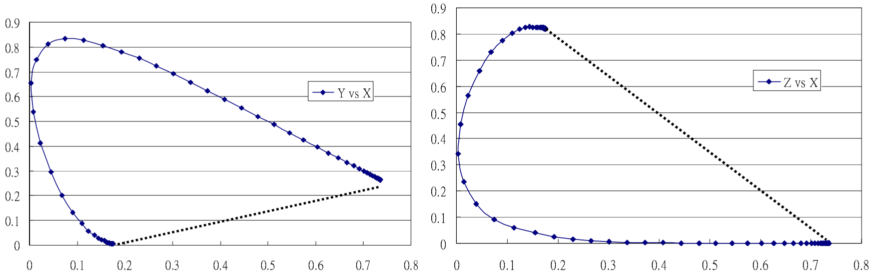
The tristimulus value, nominal tristimulus value, and chromaticity diagrams were calculated and plotted as shown in Figs. 3, 4 and 5, and are close to the standard CIE 1931 results. In order to verify the validity, a number of cases for white-points under different color temperature are illustrated in Table 2. Notice that the axes without units mean normalized value.



(a) tristimulus value for xyz

(b) normalized value for XYZ

Fig. 3. The tristimulus function versus wavelength for the RGB colors



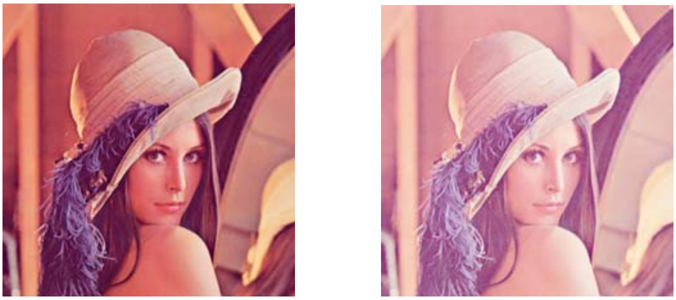
(a) Y-Z

b) Z-X

Fig. 4. The chromaticity diagram for Y-X and Z-X

Table 2. The XYZ and RGB value for the white-points for different standards calculated from the proposed algorithm

standard gamma	X	Y	Z	R_{8bit}	G_{8bit}	B_{8bit}
D65 Gamma 1.0	0.3127	0.329	0.3583	72	70	70
D65 Gamma 2.2	0.3127	0.329	0.3583	148	155	154
D50 Gamma 1.0	0.3457	0.3505	0.3038	97	72	54
D50 Gamma 2.2	0.3457	0.3505	0.3038	167	148	133
D93 Gamma 1.0	0.283	0.297	0.42	52	63	87
D93 Gamma 2.2	0.283	0.297	0.42	131	141	160
sRGB Gamma 1.0	0.313	0.329	0.358	72	70	69
sRGB Gamma 2.2	0.313	0.329	0.358	148	147	146



(a) Gamma 1.0 (b) Gamma 2.2

Fig. 5. Gamma modification under full-color YUV122 color system

7 Discussion

1) The proposed algorithm is not the same as the conventional matrix operation. Data-dependency decomposition is used to speedup the calculation in the pre-processing stage. Only three simple arithmetic operations are required. Unlike the matrix operation that processes in SAC compiler, the proposed algorithm for the color space transformation is actually a pre-processing stage before SAC compiler. No matrix data type is declared in the algorithm. Only integer data type is used. The processing for matrix operation is different from the one for the integer arithmetic operation.
2) The definition of the proposed algorithm is clear. The proposed algorithm is a pre-processing before compiled by SAC compiler. The pre-processing is helpful for speedup of the execution time. At the pre-processing stage, the three independent equations are separated independently to be suitable for parallel operation.
3) Each column of Table 1 compares the execution time under the same percent cycle and percent flip-flops. Since the color space transformation is modeled by matrix operation with the conventional form. Matrix operation is required to perform the calculation. Before using SAC compiler, this paper develops the formulation to be suitable for only three simple arithmetic operations. The implementation of three simple arithmetic operations is required in the pre-processing stage. The Conventional matrix operation is unable to speedup the color space transformation.

8 Conclusion

This paper has successfully proposed an intelligent decoupled SVD algorithm for the color space transformation of computer vision. The decoupled computation has been also implemented into the data structure suitable for the SA-C circuit-level parallel operation. With the DFG embedded in SA-C, the required VHDL codes can be further obtained. Results show that the objectives of the speedup can be achieved. The color space transformation for chromaticity has been successfully implemented.

Acknowledgment

Financial support from the National Science Council project is appreciated. The authors wish to thank Chi-Lin Tech. Inc. for providing the testing equipment. Thanks to Tzeng Tseng for typing this article kindly.

References

1. Villarreal, J.: Improving Software Performance with Configurable Logic. J. Design Automation of Embedded Systems, 325–339 (2002)
2. Li, Y.: Hardware-Software Co-Design of Embedded Reconfigurable Architectures. In: Proc. Design Automation Conf. (DAC 2000), pp. 507–512. ACM Press, New York (2000)
3. Bohm, W.: Mapping a Single Assignment Programming Language to Reconfigurable Systems. J. Supercomputing 21(2), 117–130 (2002)
4. Rinker, R.: An Automated Process for Compiling Dataflow Graphs into Hardware. IEEE Trans. VLSI, 130–139 (2001)
5. Bohm, W.: Compiling ATR Probing Codes for Execution on FPGA Hardware. In: Proc. IEEE Symp. Field-Programmable Custom Computing Machines (FCCM 2002), pp. 301–302. IEEE CS Press, Los Alamitos (2002)
6. Hammes, J., Rinker, R., Najjar, W., Draper, B.: A high-level algorithmic programming language and compiler for reconfigurable system. In: The International Engineering of Reconfigurable Hardware/Software Objects Workshop (2000)
7. Rinker, R., Carter, M., Patel, A., Chawathe, M., Ross, C., Hammes, J., Najjar, W., Bohm, W.: An automated process for compiling dataflow graphs into reconfigurable hardware. IEEE Transactions on VLSI Systems 9, 130–139 (2001)

An Efficient Method of Vehicle License Plate Detection Based on HSI Color Model and Histogram

Kaushik Deb, Heechul Lim, Suk-Ju Kang, and Kang-Hyun Jo

Dept. of Electrical, Electronics and Information Systems, University of Ulsan,
680 - 749 San 29, Mugeo 2-dong, Nam-ku, Ulsan, Korea
{debkaushik99,hclim,sjkang,jkh2009}@islab.ulsan.ac.kr

Abstract. Detecting license plate (LP) is a crucial and inevitable component of the vehicle license plate recognition (VLPR) system. In this paper to select automatically statistical threshold value in HSI color model is proposed. The proposed vehicle license plate detection (VLPD) method consists of two main stages. Initially, HSI color model is adopted for detecting candidate regions. According to different colored LP, these candidate regions may include LP regions; geometrical properties of LP are then used for classification. The proposed method is able to deal with candidate regions under independent orientation and scale of the plate. Finally, the decomposition of candidate regions contains predetermined LP alphanumeric characters by using position in the histogram to verify and detect vehicle license plate (VLP) region. Experimental results show that the proposed method is very effective in coping with different images under the variety of conditions such as complex scenes, illumination changing, distances and varied weather etc.

Keywords: Vehicle license plate detection (VLPD), HSI color model and position histogram.

1 Introduction

The task of recognizing specific object in an image is one of the most difficult topics in the field of computer vision or digital image processing. VLPD task is quite challenging from vehicle images due to the diversity of plate formats and the nonuniform outdoor illumination conditions during image acquisition. The VLPD is widely used for detecting speeding cars, security control in restricted areas, unattended parking zone, traffic law enforcement and electronic toll collection. With the rapid development of highway and the wide use of vehicle, people start to pay more and more attention on the advanced, efficient and accurate Intelligent Transportation Systems (ITS). Recently the necessity of VLPR has increased significantly. The LP detection is an important research topic of VLPR system. One of the major problems in LP detection is determining LP systems. This system must guarantee robust detection under various weather and lighting conditions, independent of orientation and scale of the plate.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 66–75, 2009.

As far as detection of the plate region is concerned, researchers have found many methods of locating license plate. For example, survey paper [1], offers to researchers a link to a public image database to define a common reference point for VLPR algorithmic assessment. In addition, this survey paper discusses about current trends and anticipated research in VLPR system. A region-based license plate detection method was presented in [2], which firstly applies a mean shift procedure in spatial-range domain to segment a color vehicle image in order to get candidate regions. In [3], a cascade framework, utilizing plate characteristics and developing fast one pass algorithms are used for a real time plate recognition system. A method based on image segmentation technique named as sliding windows (SW) was also proposed for detecting candidate region (LP region) in [4]. Image enhancement and sobel operator to extract out vertical edges and finally search plate region by a rectangular window was presented in [5]. Fuzzy logic has been applied in detecting license plates. Authors made some intuitive rules to describe the license plates and gave some membership functions for fuzzy sets e.g. "bright," "dark," "bright and dark sequence," "texture," "yellowness" to get the horizontal and vertical plate positions [6]. A technique based on extract candidate regions by finding vertical and horizontal edges from vehicle region; had also been proposed and this segmentation method is named as sliding concentric windows. Finally, vehicle license plate is verified and detected by using HSI color model and position histogram respectively [7]. Currently, some researchers prefer a hybrid detection algorithm, where license plate location method based on corner detection, edge detection, characteristics of license shape, character's connection and projection is presented in [8, 9] and an approach using mathematical morphology method to detect license plate area [10] have also been proposed.

The focus of this paper is on the consolidation of a new method to select automatically statistical threshold value in HSI color model for detecting candidate regions. These candidate regions may include LP regions; geometrical properties of LP are then used for classification. Additionally, the proposed method is able to deal with plates (candidate regions) under independent orientation and scale of the plate. More than one license plate can be detected in the same image. Finally, decomposing candidate region which contains predetermined LP alphanumeric character by using position in the histogram to verify and detect vehicle license plate region is performed.

The rest of this paper is organized as follows. The next section using the specific features of Korean VLP to be considered have been described. The three primary stages of the proposed VLPD algorithm, i.e. color segmentation, labeling and filtering and candidate decomposition of candidate region, have been discussed in details in Section 3. Finally, experimental results and some conclusions are reported in Section 4.

2 Specific Features of Korean VLP

In this section, the color arrangement of the plate and outline of the Korean VLP that are considered in this study are discussed.

Table 1. Styles of license plates

Vehicle type	Plate color	Character color
Private automobile	White	Black
	Green	White
Taxi, truck and bus	Yellow	Deep blue
Government vehicle	Yellow	Black

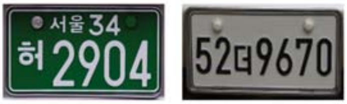


Fig. 1. Outline of the Korean license plate

Color arrangement of the plate: Korean license plates is well classified as shown in Fig. 1. Each style has a different plate color and/or character color. However, in all only five distinct colors like white, black, green, yellow, and deep blue are used in these license plates. It is worth paying attention to three different plate colors while searching for LP in an input image. Other type of vehicles, such as diplomatic cars and military vehicles, are not addressed since they are rarely seen.

Outline of the Korean VLP: Standard LP contains Korean alphabets and numbers which are shown in Fig. 1. Few LP contains Korean alphabets and numbers in two rows, in future this kind of LP is supposed to be converted into a single row. Where plate color is white and character color is black, they contain seven alphanumeric characters written in a single line. Fig. 1 shows, where plate color is green and yellow then character color is white and black, respectively, they contain Korean LP in two rows. When plate color is yellow, some LP contains all alphanumeric characters written in a single line.

3 Proposed Algorithm

The proposed VLPD algorithm consists of three distinct parts which is shown in Fig. 2. The first one deals with by using HSI color model the detection of the candidate region, i.e., the license plate. The second part allows procedures for refining candidate region by using labeling and filtering. According to different colored LP these candidate regions may include rectangular LP regions; geometrical properties of LP such as area, bounding box, and aspect ratio are then used for classification. In this part also includes operations for determine the angle of the candidate - rotation adjustment. Measurement such as center of area and the least second moment are employed to solve the rotation adjustment. The third part includes performances for candidate's decomposition. Finally, decomposing of candidate region contains predetermined LP alphanumeric character by using position in the histogram to verify and detect vehicle license plate (VLP) region.

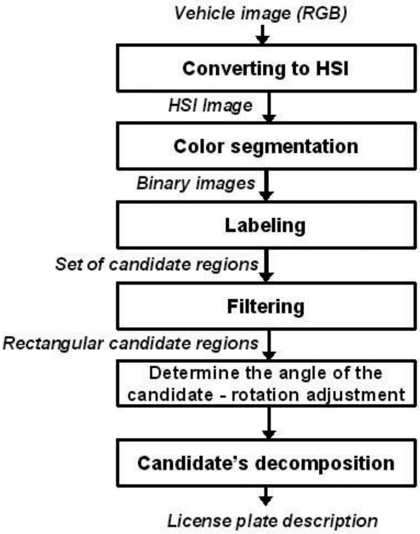

Vehicle image (RGB)
Converting to HSI
HSI Image
Color segmentation
Binary images
Labeling
Set of candidate regions
Filtering
Rectangular candidate regions
Determine the angle of the candidate - rotation adjustment
Candidate's decomposition
License plate description

Fig. 2. Main algorithm scheme for detecting license plate region

3.1 Color Segmentation

In the proposed method, input vehicle images are converted into HSI color images. Then the candidate regions are found by using HSI color model on the basis of using hue, saturation and/or intensity. Many applications use the HSI color model. Machine vision uses HSI color model in identifying the color of different objects. Plate color information is used to detect candidate regions in our experiments, and shape properties of LP allow reducing number of LP-like candidates. One of the common ways of color based vehicle license plate detection can be formalized as follows:

$$R(x,y) > \alpha_R;\ G(x,y) > \alpha_G;\ B(x,y) > \alpha_B \tag{1}$$

$$R(x,y) - G(x,y) > \beta_{RG};\ R(x,y) - B(x,y) > \beta_{RB} \tag{2}$$

where R, $G\ and\ B$ are red, green and blue components of $x \times y$ image. α and β are predefined coefficients. Equation (1) sets up limitations for the minimal values of pixel components. Equation (2) formalizes dependencies between pixel components for LP. Generally, common way of using color based vehicle license plate detection was based on two types of restrictions: first restriction is based on equations (1) and (2). It provides good results in good lighting conditions. However, it is not good for low-contrast images. Pixel belongs to green and yellow LP, respectively like following equations (3) and (4)

$$b_{green} = \begin{cases} 1, [\{R(x,y) \leq\ 0.85{\cdot}G(x,y)\}\ \&\ \{B(x,y) \leq\ 0.90{\cdot}G(x,y)\}] \\ 0, \qquad\qquad\qquad\qquad otherwise \end{cases} \tag{3}$$

$$b_{yellow} = \begin{cases} 1, [\{B(x,y) \leq\ 0.90{\cdot}R(x,y)\}\ \&\ \{B(x,y) \leq\ 0.80{\cdot}G(x,y)\}] \\ 0, \qquad\qquad\qquad\qquad otherwise \end{cases} \tag{4}$$

where $b_{green}\ and\ b_{yellow}$ are green and yellow candidate binary masks. The second restriction is based on equations (3) and (4), and a threshold value is taken heuristically. It provides stable result whereas in bad lighting condition it is too sensitive.

In this proposed method, LP detection is based on its color properties, namely mean and standard deviation values of hue. For detection green and yellow LP pixels, hue parameter of HSI color is used in our experiment. To detect white LP pixels hue value is meaningless, hence only saturation and intensity parameters are important for this case. To estimate these properties, we used 30 images of LP taken under different lighting and weather conditions. After training from those sample data, the mean and standard deviation values of hue are computed for detection of green and yellow LP pixels. Detecting white license plate pixels, the mean and standard deviation values of saturation and intensity are computed to detect green, yellow and white LP from vehicle images.

Detection of green and yellow LP pixels the binarization process can be formulated as follows:

$$b_{green} = \begin{cases} 1,\ \ [\{\mu^H - \sigma^H\ \leq\ H(x,y)\ \leq\ \mu^H + \sigma^H\}\ \&\ \{S(x,y) \geq 0.08\} \\ \qquad\qquad\qquad \&\ \{0.05\ \leq\ I(x,y)\ \leq\ 0.95\}] \\ 0, \qquad\qquad\qquad otherwise \end{cases} \tag{5}$$

$$b_{yellow} = \begin{cases} 1,\ \ [\{\mu^H - \sigma^H\ \leq\ H(x,y)\ \leq\ \mu^H + \sigma^H\}\ \&\ \{S(x,y) \geq 0.12\} \\ \qquad\qquad\qquad \&\ \{0.20\ \leq\ I(x,y)\ \leq\ 0.80\}] \\ 0, \qquad\qquad\qquad otherwise \end{cases} \tag{6}$$

where $H(x,y),\ S(x,y)\ and\ I(x,y)$ are hue, saturation and intensity components of $x^{th},\ y^{th}$ pixel respectively. $\mu^H,\ \sigma^H$ are mean hue and hue standard deviation values for green and yellow LP of sample data, respectively. However, the automatic focus and white balancing of camera often cause the changing illumination. Our proposed LP detection method can work well in normal illumination condition, but it seems not good enough to work in bad illumination conditions. To overcome this problem, we use an adaptive LP detection method; we use it in the case of really high or low illumination condition. For normal, low and high illumination conditions white license plate pixels the binarization process can be formulated as follows, respectively:

$$b_{white(n)} = \begin{cases} 1,\ \ [S(x,y)\ \leq\ (\mu^S + \sigma^S)\ \&\ I(x,y)\ \geq\ (\mu^I + 0.25 \cdot \sigma^I)] \\ 0, \qquad\qquad\qquad otherwise \end{cases} \tag{7}$$

$$b_{white(l)} = \begin{cases} 1,\ \ [S(x,y)\ \leq\ (\mu^S + \sigma^S)\ \&\ I(x,y)\ \geq\ (\mu^I - 0.33 \cdot \sigma^I)] \\ 0, \qquad\qquad\qquad otherwise \end{cases} \tag{8}$$

$$b_{white(h)} = \begin{cases} 1, & [S(x,y) \ \leq \ (\mu^S + \sigma^S) \ \& \ I(x,y) \ \geq \ (\mu^I + 0.50 \cdot \sigma^I)] \\ 0, & otherwise \end{cases} \tag{9}$$

where $S(x,y)$, $I(x,y)$, are saturation and intensity components of x^{th}, y^{th} pixel respectively. μ^S, μ^I are mean values for saturation and intensity, σ^S, σ^I are standard deviation values for saturation, intensities of white LP of sample data, respectively. $b_{white(n)}$, $b_{white(l)}$ and $b_{white(h)}$ are white candidate binary masks. Color segmentation parameters are very sensitive in order to detect as much candidates as possible. All false candidates will be filtered out on the next stages. After the segmentation, there may still exist noises such as small holes or/and small bulges of the target candidate regions. These problem may be resolved by using mathematical morphology closing operation which is dilation followed by erosion to fill in holes and gaps smaller than the structuring element on the plate image. Removing those holes play an important role while calculating bounding box region.

3.2 Labeling and Filtering

After the candidate regions are obtained by applying color segmentation, features of each region are to be extracted in order to correctly differentiating the LP regions from others. Next step of proposed algorithm is labeling the connected components. In the proposed method, a recursive algorithm is implemented for connected components labeling operation. Recursive algorithm works on one component at a time, but can move all over the image. On this step we extract candidate regions which may include LP regions from the binary mask obtained on the previous step. During this step main geometrical properties of LP candidate such as area, bounding box and aspect ratio are computed.

A bounding box is a rectangle whose horizontal and vertical sides enclose the region and touches its topmost, bottommost, leftmost and rightmost points. Rectangularity is defined as the ratio of the area of candidate object's MER (minimum enclosing rectangle) and the area of the object. Here, the area measured in pixels and indicate the relative size of the object. The aspect ratio (also called elongation or eccentricity), is defined by the ratio of the bounding box of an object. This can be found by scanning the image and the minimum and maximum values on the row and the columns, where the object lies. This ratio is defined by:

$$\rho_{\mathbf{A}} \ = \ \frac{c_{max} - c_{min} + 1}{r_{max} - r_{min} + 1} \tag{10}$$

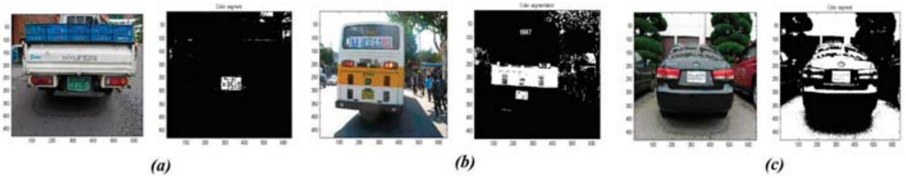


Fig. 3. Vehicle image LP and its color segmentation results using HSI color model

where c and r indicate columns and rows, respectively. Those objects whose measurements fulfill the criteria as follows: $\rho_{\mathbf{A}} = (1.0, 3.0)$ for green LP, $\rho_{\mathbf{A}} = (1.0, 2.0)$ for yellow LP, $\rho_{\mathbf{A}} = (1.0, 6.0)$ for white LP are considered as candidate plate regions. These parameters are used filtering operation to eliminate LP-like objects from candidate list. Filtering operation is done on geometrical properties of LP regions. As license plates can appear at many different angles to the cameras optical axis, each rectangular candidate regions is rotated until they are all aligned in the same way before the candidate decomposition. Following the successful filtering operation in image, measurement such as center of area and the axis of least second moment are employed to solve the rotation adjustment problem.

The least second moments provides the principal axis as the orientation with the candidate object. For getting principal axis of detected candidate region, we compute central moments of detected candidate region. The central moments are defined as:

$$\mu_{pq} = \sum_{r=0}^{N-1} \sum_{c=0}^{N-1} (r - \overline{r})^p (c - \overline{c})^q I(r, c) \tag{11}$$

where $\overline{r}$ and $\overline{c}$ are the centroid for the candidate object. We apply this result to obtain a direction of principal axis by centroid of detected candidate region. Angle of principal axis moments is obtained as

$$\theta \ = \ \frac{1}{2} arctan(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}) \tag{12}$$

where θ denotes an angle between basis horizontal coordinate and principal axis of region.

3.3 Candidate Decomposition

Information extracted from image and intensity histograms plays a basic role in image processing, in areas such as enhancement, segmentation and description. In this section, verification and detection of the VLP region as well as character segmentation are considered and discussed in this study. Once the candidate area is binarized the next step is to extract the information. At first, regions

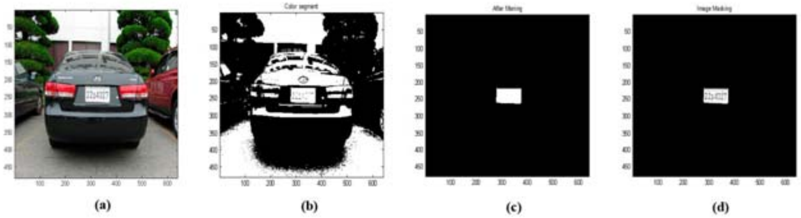


Fig. 4. Steps for license plate segmentation: (a) initial image, (b) result of color segmentation, (c) detected candidate after filtering, and (d) plate detection

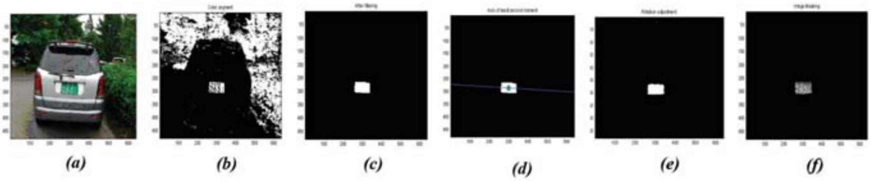


Fig. 5. Steps for license plate segmentation (green background LP): (a) input image, (b) color segmentation result, (c) detected candidate after filtering, (d) principal axis, (e) rotation adjustment, and (f) candidate region detection

without interest such as border or some small noisy regions are eliminated; the checking is made by height comparison with other plate characters height.

Fig. 7 shows the results for verifying predetermined alphanumeric character (green LP): (a) extracting candidate region, (b) horizontal position histogram without LP border, (c) processing of upper row after normalization, (d)processing of lower row after normalization, and (e) character extraction; For (white LP): a) extracting candidate region, (b) vertical position histogram with LP border, (c) horizontal position histogram with LP border, (d) horizontal position histogram without LP border,(e) view of normalization candidate region after removing border and noisy area, (f) vertical position histogram (seven peaks for predetermined seven alphanumeric characters in LP region), and (g) character extraction.

4 Experimental Results and Conclusions

All experiments have been done on Pentium-IV 2.4 MHz with 1024 MB RAM under Matlab environment. In the experiments, 150 images were used the size is 640*480 pixels, some sampled images are shown in Fig. 6. The images are taken from (a) different illuminations (strong sunshine, shadow and night time), (b) complex scenes where several objects such as trees, light post in front of vehicles, (c) various environments in campus parking and access areas and (d) damaged LP as bent or old. They were taken in distance from 2 up to 7 meters and the camera was focused in the plate region. Under these conditions, success of LP detection has reached to more than 92%. A common failure of the proposed VLPD system is the failure to detect the boundaries or border of license plates. This occurs when vehicle bodies and their license plate possess similar colors. The

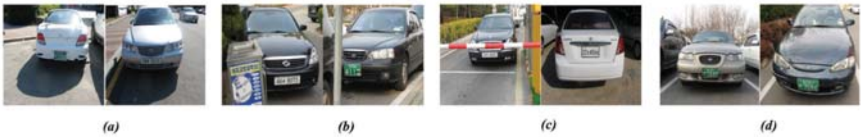


Fig. 6. Example images: (a) different illuminations, (b) complex scenes, (c) various environments and (d) damaged license plates

Table 2. Required average computation time

Stage	Avg time (sec)	Std. deviation (sec)
Color segmentation	0.16	0.07
Filtering (bounding box and aspect ratio)	0.07	0.02

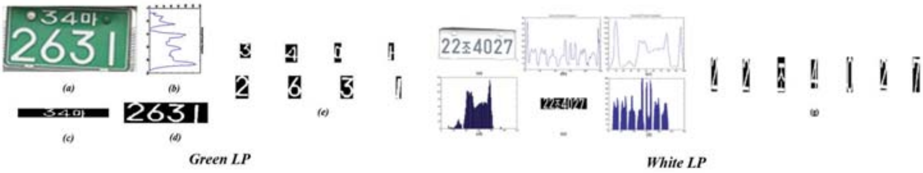


Fig. 7. Steps for verifying predetermined alphanumeric character (green and white LP)

average computational time for the color segmentation and filtering operations of the proposed method are shown in Table 2.

Conclusion, a new method is adopted in this paper to select automatically statistical threshold value in HSI color model. In the proposed method candidate regions are found by using HSI color model. These candidate regions may include LP regions; geometrical properties of LP are then used for classification. The proposed method is able to deal with plates (candidate regions) under independent orientation and scale of the plate. More than one license plate can be detected in the same image. Finally, VLP regions containing predetermined LP alphanumeric character are verified and detected by using position histogram. Color arrangement and predetermined LP alphanumeric character of the Korean license plate is an important feature for verification and detection of license plate regions.

From the experiment, different illumination conditions, complex scenes, varied distances between vehicle and camera often occurred. In the case, the result that has been confirmed is very much effective when the proposed approach is used. However, the proposed method is sensitive when vehicle bodies and their license plates possess similar colors. We leave these issues to be considered in future studies.

Acknowledgments. The authors would like to thank to Ulsan Metropolitan City, KOTEF, MKE and MEST of Korean Government which partly supported this research through the NARC and post BK21 project at University of Ulsan. Also, we express special thanks to IITA for their international graduate students scholarship.

References

1. Anagnostopoulos, C., Anagnostopoulos, I., Loumos, V., Kayafas, E.: License plate-recognition from still images and video sequences: A survey. IEEE Trans. Intell. Transp. Syst. 9(3), 377–391 (2008)

2. Jia, W., Zhang, H., He, X.: Region-based License Plate Detection. J. Network and comput. Applications 30(4), 1324–1333 (2007)
3. Wang, S., Lee, H.-J.: A cascade framework for a real-time statistical plate recognition system. IEEE Trans. Inf. Forensics Security 2(2), 267–282 (2007)
4. Anagnostopoulos, C., Anagnostopoulos, I., Loumos, V., Kayafas, E.: A License Plate-Recognition Algorithm for Intelligent Transportation System Applications. IEEE Trans. Intell. Transp. Syst. 7(3), 377–392 (2006)
5. Zheng, D., Zhao, Y., Wang, J.: An Efficient Method of License Plate Location. Pattern Recognit. Lett. 26(15), 2431–2438 (2005)
6. Chang, S., Chen, L.-S., Chung, Y.-C., Chen, S.-W.: Automatic License Plate Recognition. IEEE Trans. Intell. Transp. Syst. 5(1), 42–53 (2004)
7. Deb, K., Chae, H.-U., Jo, K.-H.: Parallelogram and Histogram based Vehicle License Plate Detection. In: IEEE ICSMA, pp. 349–353. IEEE Press, New York (2008)
8. Xu, Z., Zhu, H.: An Efficient Method of Locating Vehicle License Plate. In: IEEE ICNC, pp. 180–183. IEEE Press, New York (2007)
9. Zhang, C., Sun, G., Chen, D., Zhao, T.: A Rapid Locating Method of Vehicle License Plate based on Characteristics of Characters Connection and Projection. In: IEEE Conf. on Industrial and Applications, pp. 2546–2549. IEEE Press, New York (2007)
10. Martin, F., Garcia, M., Alba, J.L.: New Methods for Automatic Reading of VLP's (Vehicle License Plates). In: IASTED Int. Conf. on SPPRA (2002)

A Robust Method for Automatically Detecting Cracks on Noisy Concrete Surfaces

Yusuke Fujita and Yoshihiko Hamamoto

Graduate School of Medicine, Yamaguchi University
2-16-1 Tokiwadai, Ube, 755-8611, Japan

Abstract. Automatic crack detection from concrete surface images is very effective for nondestructive testing. In our previous work, we proposed two preprocessing methods for automatic crack detection from noisy concrete surfaces. In this paper, we propose an automatic crack detection method after the preprocessings. The proposed method consists of two steps. One is relaxation process to prevent noises, and the other is a improved locally adaptive thresholding to detect cracks exactly. We evaluate the performance of the proposed method using 60 actual noisy concrete surface images, compared to those of the conventional thresholding techniques. Experimental results show that the robustness and the accuracy of the proposed method are good.

1 Introduction

Automatic crack detection from concrete surface images is very effective for nondestructive testing. However, in general, it is very difficult to detect cracks automatically from noisy concrete surfaces. Because these images include various blebs, stains, shadings and irregularities in crack shape and size. It is necessary to overcome these problems, in order to detect cracks automatically from noisy concrete surfaces with high accuracy.

Recently, some methods for crack detection by means of visual inspection have been proposed [1,2,3,4,5]. Ito et al. proposed an inspection system to detect and analyze cracks on the concrete surfaces [1]. Some image processing techniques, such as wavelet transform, shading correction and binarization, are combined to extract cracks from concrete surfaces. However, in addition to original images, corrective images are needed for shading correction. Abdel-Qader et al. compared the effectiveness of crack detection among wavelet transform, Fourier transform, Sobel filter, and Canny filter [2]. Hutchinson et al. used Canny filter and wavelet transform for crack detection [3]. However, it is not considered varying crack width in a concrete surface image. Yamaguchi et al. proposed a crack detection method based on percolation model [4]. Binarization is included after percolation processing. However, optimization of threshold in that method is not considered. Kawamura et al. proposed use of a genetic algorithm for the semi-automatic optimization of combining of the image processing techniques or choice of the parameters [5]. These conventional methods need to choose parameters or the

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 76–85, 2009.

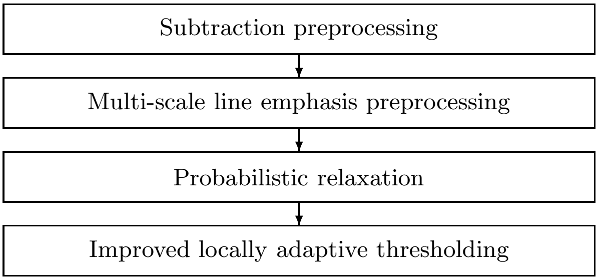

Subtraction preprocessing
Multi-scale line emphasis preprocessing
Probabilistic relaxation
Improved locally adaptive thresholding

Fig. 1. Flow of the proposed method

combination of processes very carefully by operators. Automation is needed for more effective and objective visual inspection.

In our previous work, we proposed two preprocessing methods for noisy concrete surface images [6]. In this paper, we propose a robust automatic detection method after two preprocessings. The proposed method consists of two steps. One is relaxation process to prevent noises and the other is a improved locally adaptive thresholding to detect cracks exactly. We evaluate the performance of the proposed method using 60 actual noisy concrete surface images. Experimental results show that the robustness and the accuracy of the proposed method are good.

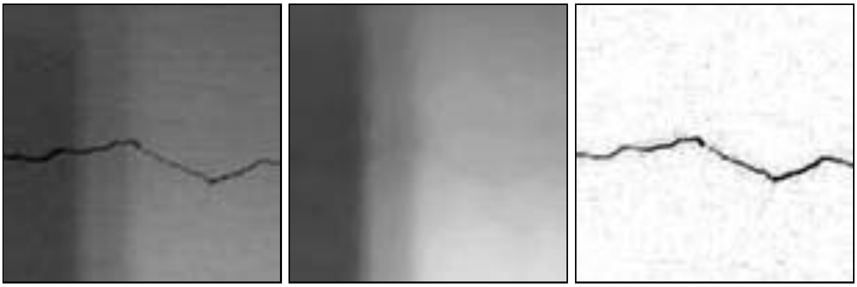
2 Outline and Preprocessings

2.1 Crack Detection Procedure

In practical concrete surface images, there are many problems such as shading, blebs and stains, as well as cracks, to complicate crack detection. For accurate and automatic crack detection, it is necessary to overcome these problems. In our previous work, we proposed two preprocessings. The first preprocessing is subtraction process with the median filter using only an original image to prevent influence of shadings. After subtraction step, multi-scale line emphasis process with the Hessian matrix is used to extract line structures such as cracks varying in widths. After two preprocessings, the proposed method including probabilistic relaxation, for detecting cracks and preventing noises, and the improved locally adaptive thresholding, for detecting cracks more finely, are used to detect cracks. Fig. 1 shows the flow of the proposed method.

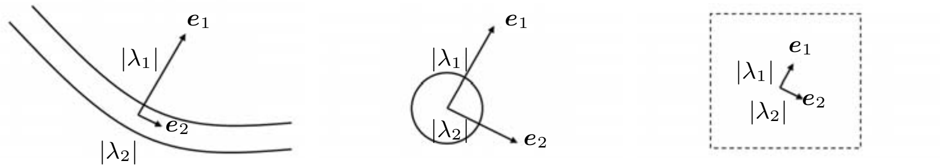
2.2 Preprocessings

Subtraction Preprocessing. Fig. 2 illustrates a removal of slight variation like shadings. At first, we use the median filter to smooth the original image to remove cracks. The original image is subtracted from the smoothed image to obtain a subtracted image as the output of the first preprocessing. The subtracted image is obtained by



(a) (b) (c)

Fig. 2. Removal of slight variation by the subtraction preprocessing. (a) Original image. (b) Smoothed image with the median filter. (c) Subtracted image, which is inverted in intensity level.



line structure $|\lambda_2| \gg |\lambda_1| \approx 0$ (a)

blob structure $|\lambda_1| \approx |\lambda_2| \gg 0$ (b)

sheet structure $|\lambda_1| \approx |\lambda_2| \approx 0$ (c)

Fig. 3. (a), (b) and (c) show eigenvalues and eigenvectors of the Hessian matrix in a line structure, a blob structure and a sheet structure, respectively. λ_1 and λ_2 mean the eigenvalues, and their corresponding eigenvectors are $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, respectively. The vectors indicate direction of eigenvectors, and the length of vector corresponds to the magnitude of the eigenvalue.

$$I'(\boldsymbol{x}_i) = \max \begin{cases} \text{median}_{\boldsymbol{x}_j \in R_i} I(\boldsymbol{x}_j) - I(\boldsymbol{x}_i) \\ 0 \end{cases} , \qquad (1)$$

where $I(\boldsymbol{x})$ denotes the intensity of the pixel $\boldsymbol{x}$, and R_i denotes the neighborhood of the pixel $\boldsymbol{x}_i$.

Multi-scale Line Emphasis Preprocessing. The line filter based on the Hessian matrix [7] is used to measure similarity of a bright line, in order to emphasize cracks and to remove noises such as blebs or stains. The Hessian matrix has the partial second derivatives of the image, and the Hessian matrix of an image $I(\boldsymbol{x})$ is given by

$$\nabla^2 I(\boldsymbol{x}) = \begin{bmatrix} I_{xx}(\boldsymbol{x}) & I_{xy}(\boldsymbol{x}) \\ I_{yx}(\boldsymbol{x}) & I_{yy}(\boldsymbol{x}) \end{bmatrix} , \qquad (2)$$

where partial second derivatives of the image $I(\boldsymbol{x})$ are represented by expressions as $I_{x^i y^j}(\boldsymbol{x}) = \frac{\partial^2}{\partial x^i \partial y^j}, (i + j = 2)$. The combination of two eigenvalues of the matrix can express the difference in structure among a line, a blob or a sheet, in the image. Fig. 3 illustrates the eigenvalues and the eigenvectors of the Hessian matrix in a line structure, a blob structure and a sheet structure. Let λ_1 and λ_2 to be the eigenvalues. When $|\lambda_2| \gg |\lambda_1| \approx 0$, the structure should be regarded as a line structure in the 2-D space. When the value of λ_1 is positive, the structure involves concavity in the estimated line direction. Therefore, we derived a generalized measure of similarity to a line as,

$$\lambda_{12} = \begin{cases} |\lambda_2| \left(1 + \frac{\lambda_1}{|\lambda_2|}\right) = |\lambda_2| + \lambda_1 \\ \qquad\qquad\qquad if\ \lambda_2 \leq \lambda_1 \leq 0 \\ |\lambda_2| \left(1 - \alpha\frac{\lambda_1}{|\lambda_2|}\right) = |\lambda_2| - \alpha\lambda_1 \\ \qquad\qquad if\ \lambda_2 < 0 < \lambda_1 < \frac{|\lambda_2|}{\alpha} \\ 0 \qquad\qquad\qquad otherwise\,, \end{cases} \tag{3}$$

where $0 \leq \alpha \leq 1$.

In the implementation, instead of computing directly the second order derivative of the image, the Gaussian convolution is combined with the second derivative in Eq. (4) in order to turn the filter response to the specific widths of lines as well as to reduce the effect of noises.

$$I_{x^i y^j}(\boldsymbol{x}, \sigma_f) = \left\{ \frac{\partial^2}{\partial x^i \partial y^j} G(\boldsymbol{x}; \sigma_f) \right\} * I(\boldsymbol{x}),\ (i + j = 2), \tag{4}$$

where $G(\boldsymbol{x}; \sigma_f)$ is an isotropic Gaussian function with standard deviation σ_f, and $*$ denotes the convolution. Then we select the maximum response among the multiple scales. The multi-scale integration of the filter responses is defined by

$$R(\boldsymbol{x}) = \max_{\sigma_f} \sigma_f^2 \lambda_{12}\left(\boldsymbol{x}; \sigma_f\right), \tag{5}$$

where $\lambda_{12}\left(\boldsymbol{x}; \sigma_f\right)$ is a filter response based on the second derivatives of Gaussian with σ_f. We normalize the filter responses of each scale with σ_f. Let $\sigma_f = s^{f-1}\sigma_1$ $(f = 1, 2, \ldots, n)$ be a discrete sample of σ_f, where σ_1 is the minimum scale and s is a scale factor determining the sampling interval of σ_f.

3 Detection Processing

Binarization is important for automatic crack detection after the above preprocessings. Many threshold selection techniques for binarization have been proposed [8,9,10]. However, the global thresholding method may make pieces of unconnected crack instead of over-detection, or it may generate noises, not crack, to detect cracks completely. In the case of locally adaptive thresholding techniques, a local threshold for each pixel is examined in its neighborhood. It is difficult to assess whether cracks exist in each local area or not, in preventing noises with use of the method.

3.1 Probabilistic Relaxation Based Line Structure

Probabilistic relaxation is mainly used as one of optimization techniques for solving labeling problem [11,12]. By probabilistic relaxation, it is possible to label the objects to be extracted from noisy data.

For each pixel, the probability $P(\lambda_c)$ corresponding to crack is assigned to the logarithmic transformation of the output of the preprocessings, as the initial probability. Here, we used a logarithm transformation as,

$$P_i(\lambda_c) = \log(R(\boldsymbol{x}_i) + 1)/\log(R_{max} + 1), \tag{6}$$

where $R(\boldsymbol{x}_i)$ means the output of pixel $\boldsymbol{x}_i$ and R_{max} is the maximum value in the image. The other hand, the probability corresponding to background is assigned to $P_i(\lambda_b) = 1 - P_i(\lambda_c)$, where λ_c and λ_b denote label for crack and background region, respectively.

Updating the probabilities of labels is done by considering the probabilities of labels for neighborhood. The updating rule proposed by Peleg [12] is

$$P_i'(\lambda_k) = \frac{P_i(\lambda_k)Q_i(\lambda_k)}{\sum_{l \in L} P_i(\lambda_l)Q_i(\lambda_l)}. \tag{7}$$

The non-ambiguity $Q_i(\lambda_k)$ is regarded as the strength of support that label λ_k at pixel $\boldsymbol{x}_i$ gets from its all neighbor pixels $\boldsymbol{x}_j \in R_i$. For example, it is defined simply as follows:

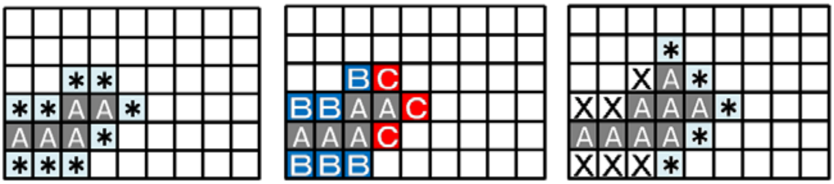
$$Q_i(\lambda_k) = \frac{1}{N(R_i)} \sum_{\boldsymbol{x}_j \in R_i} P_j(\lambda_k), \tag{8}$$

where $N(R)$ denotes the number of pixels in the region R. Using Eq. (8), $Q_i(\lambda_k)$ is the average of estimates for label λ_k in neighborhood of $\boldsymbol{x}_i$.

However, in crack detection, if the average of estimates in neighborhood is used as non-ambiguity, line structure like crack in the image may be removed unexpectedly due to its structure property. Therefore, we divide the neighborhood to four sub-regions according to direction and estimate the non-ambiguity in each subregion, in order to remain line structures as cracks. This means that four estimates of each pixel are calculated along four different angular directions 0, $1/4\pi$, $1/2\pi$, and $3/4\pi$ passing through the pixel. The maximum value of four estimates is used to update the probability. We can update probability $P_c(\lambda_k)$ by

$$P_i'(\lambda_c) = \max_d \frac{P_i(\lambda_c)Q_i^d(\lambda_c)}{P_i(\lambda_c)Q_i^d(\lambda_c) + P_i(\lambda_b)Q_i^d(\lambda_b)} \tag{9}$$

$$Q_i^d(\lambda_k) = \frac{1}{N(R_i^d)} \sum_{\boldsymbol{x}_j \in R_i^d} P_j(\lambda_l), \tag{10}$$



(a) (b) (c)

Fig. 4. The improved locally adaptive thresholding. (a) Initial candidate region. Pixels labeled as "A" are classified to crack. Pixels labeld as "$*$" are in candiate region. (b) Result of thresholding for pixels in candidate region. Pixels labeld as "B" or "C" are classified to background or crack on the improved locally adaptive thresholding, respectively. (c) Updating candidate region from (b). Pixels labeled as "X" classifeid once to background, and these are not added to candidate region again.

where $d = 0, \pi/4, \pi/2, 3/4\pi$ means the direction and R_i^d is the subregion of the direction d. The updating process for all pixels is repeated until convergence.

3.2 Improved Locally Adaptive Thresholding

Fig. 4 illustrates our locally adaptive thresholding. The improved locally adaptive thresholding is used to close undesirable unconnected pieces of cracks and also to fill in missing parts of cracks. The neighbor pixels of cracks detected already are added to candidate region and are determined whether the pixels in the region are a piece of crack or not. Here, the local thresholding technique is used, where the threshold value is decided in its neighborhood. This approach is called the locally adaptive thresholding because the threshold value is selected for each pixel in the limited region, while thresholding isn't applied in the region where no crack exists. So this approach is considered as a improved locally adaptive thresholding technique that is able to prevent noises. In the updating candidate region, pixels in the neighborhood of the pixel labeled as cracks are added to the candidate region, while the pixels once classified as non-crack by improved local thresholding are not added to the candidate. Updating process is performed repeatedly until convergence.

4 Experimental Results

In this section, we evaluate the performance of the proposed method, compared to those of the global thresholding and the local thresholding. In this experiment, we used 60 actual concrete surface images including various blebs, brains and shadings. These digitized images with 640 × 480 pixels have approximately 0.1 mm pixel size, and 8-bit gray levels. Note that our target cracks are over 0.2 mm in width.

For comparison, three detection techniques are evaluated in terms of sensitivity, specificity and precision [13,14]. Sensitivity, specificity and precision are estimated as

$$\text{sensitivity} = \frac{\text{Cracks correctly classified}}{\text{Total cracks}} \tag{11}$$

$$\text{specificity} = \frac{\text{Backgrounds correctly classified}}{\text{Total backgrounds}} \tag{12}$$

$$\text{precision} = \frac{\text{Cracks correctly classified}}{\text{Cracks correctly classified} + \text{Backgrounds incorrectly classified}}. \tag{13}$$

The ground truth image for evaluation is created manually by a human observer.

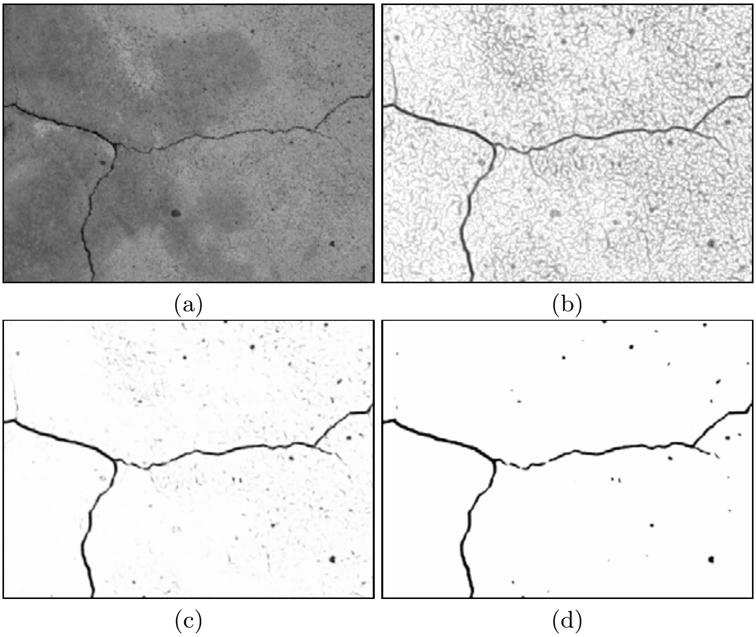
The evaluation is performed as follows. Preprocessings as described in section 2 are applied to each original image. The size of median filter is 21 × 21 pixels. Parameters of the multi-scale line emphasis filter are $\sigma_1 = \sqrt{2}$, $s = \sqrt{2}$, $n = 4$, and $\alpha = 0.25$ [6]. Three detection techniques are applied to detect cracks from the preprocessed image. Otsu's thresholding technique [15] is used as global thresholding technique. The technique selects a threshold based on integration of the gray-scale histogram to maximize the likelihood that the threshold is chosen so as to split the image between cracks and background. In the local thresholding, local threshold for each pixel is examined by Otsu's technique in its 21 × 21 neighborhood. In the improved locally adaptive thresholding of the proposed method, local threshold for each pixel is examined by Otsu's technique in its 21 × 21 neighborhood. Results of three detection techniques are compared with ground truth image, to evaluate in terms of the sensitivity, the specificity and the precision.

Averages and 95% confidence interval of sensitivity, specificity and precision from the 60 images are shown in Table 1. The proposed method and the local thresholding have higher sensitivity than the global thresholding. These two techniques work out threshold sensitive to cracks, for each pixel. The proposed method and the global thresholding have higher specificity and higher precision than the local thresholding. On the other hand, the local thresholding is sensitive to noise. The proposed method works well. Fig. 5 shows probabilistic relaxation of the proposed method. Fig. 6 shows the improved locally adaptive thresholding.

Table 1. Comparison of detection techniques on 60 images in terms of sensitivity, specificity and precision

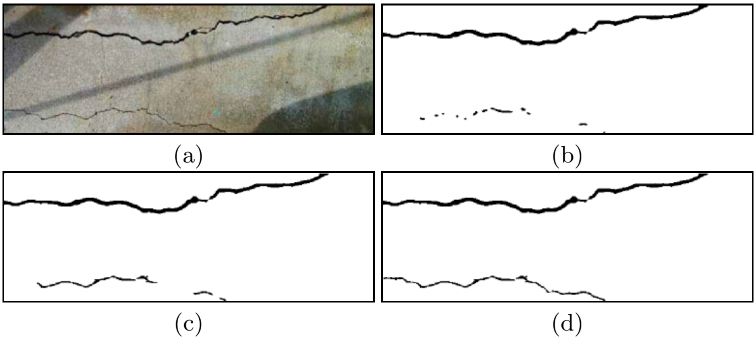
	Global thresholding	Local thresholding	Proposed method
Sensitivity	0.722	0.815	0.801
	0.685, 0.759	0.796, 0.834	0.797,0.805
Specificity	0.993	0.922	0.992
	0.992, 0.994	0.918, 0.926	0.991, 0.993
Precision	0.655	0.147	0.606
	0.624, 0.686	0.134, 0.16	0.581, 0.631

The first and second lines of the table are the average and 95% confidence interval from the 60 images, respectively.



(a) (b) (c) (d)

Fig. 5. Probabilistic relaxation process of the proposed method. (a) Original image. (b) Result of applying preprocessings (a) ($t = 0$), where t denotes the iteration number of updating in relaxation. (c),(d) Result of Probabilistic relaxation process from (b) ((c) $t = 2$, (d) $t = 10$).



(a) (b) (c) (d)

Fig. 6. Improved local thresholding of the proposed method. (a) Original image. (b) The result of relaxation. (c), (d) Results of the improved locally adaptive thresholding from (b) ((c) $t = 10$, (d) $t = 67$), where t denotes the iteration number of the local thresholding. The Updating process is converged at $t = 67$.

Experimental results suggest that probabilistic relaxation performs to limit the regions to detect cracks, and the improved locally adaptive thresholding of the proposed method is less sensitive to noise. The visually comparison of the

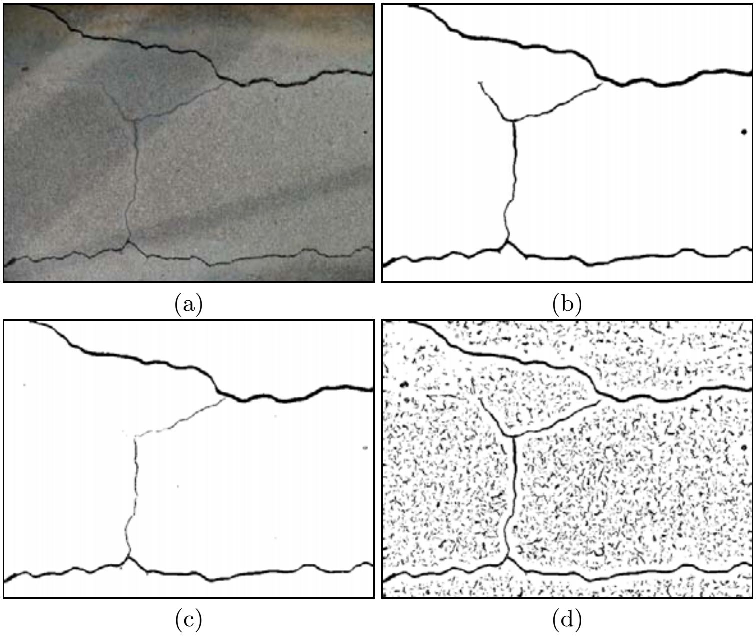

(a) (b) (c) (d)

Fig. 7. Results of crack detection. (a) Original image. (b)-(d) Results of crack detection after the preprocessings, by the proposed method, by the global thresholding and by the local thresholding, respectively.

three method after applying the preprocessings are shown in Fig 7. These results show the proposed method can detect cracks with high accuracy, compared with conventional thresholding approaches.

5 Conclusion

In this paper, we proposed a new crack detection method, in which probabilistic relaxation and the improved locally adaptive thresholding are included, in order to detect cracks automatically from noisy concrete surfaces. Probabilistic relaxation performs to prevent noises, and the improved locally adaptive thresholding is performed detect cracks exactly. We evaluated the performance of the proposed method using actual images, compared to those of conventional thresholdings. The experimental results show that the proposed method is clearly more effective than the well-known global thresholding and the local thresholding.

References

1. Ito, A., Aoki, Y., Hashimoto, S.: Accurate extraction and measurement of fine cracks from concrete block surface image. In: Proc. IECON, vol. 3, pp. 2202–2207 (2002)

2. Abdel-Qader, I., Abudayyeh, O., Kelly, M.E.: Analysis of edge detection techniques for crack identification in bridges. J. of Computing in Civil Engineering 17(3), 255–263 (2003)
3. Hutchinson, T.C., Chen, Z.: Improved image analysis for evaluating concrete damage. J. of Computing in Civil Engineering 20(3), 210–216 (2006)
4. Yamaguchi, T., Nakamura, S., Saegusa, R., Hashimoto, S.: Image-based crack detection for real concrete surfaces. Trans. on Electrical and Electronic Engineering 3, 128–135 (2008)
5. Kawamura, K., Miyamoto, A., Nakamura, H., Sato, R.: Proposal of a crack pattern extraction method from digital images using an interactive genetic algorithm. Proc. of Japan Society of Civil Engineers 742, 115–131 (2003)
6. Fujita, Y., Mitani, Y., Hamamoto, Y.: A method for crack detection on a concrete structure. In: Proc. of the 18th Int. Conf. on Pattern Recognit., pp. 901–904 (2006)
7. Sato, Y., Nakajima, S., Shiraga, N., Atsumi, H., Yoshida, S., Koller, T., Gerig, G., Kikins, R.: Three-dimensional multi-scale line filter for segmentation and visualization of curvatures in medical images. Medical Image Analysis 2(2), 143–168 (1999)
8. Weszka, J.S.: SURVEY: A Survey of threshold selection techniques. Computer Graphics and Image Process. 7, 259 (1978)
9. Lee, S.U., Chung, S.Y., Park, R.H.: A comparative study of several global thresholding techniques for segmentation. Computer Graphics and Image Process. 52, 171–190 (1990)
10. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. J. of Electronic Imaging 13(1), 146–168 (2004)
11. Rosenfeld, A., Hummel, R.A., Zucker, S.W.: Scene labeling using relaxation operation. IEEE Trans. Syst., Man, Cybern. SMC-6(6), 420–433 (1976)
12. Peleg, S.: A new probabilistic relaxation scheme. IEEE Trans. Pattern Analysis and Machine Intelligence PAMI-2(4), 362–369 (1980)
13. Fawcett, T.: An introduction to ROC analysis. Pattern Recognit. Letters 27(8), 861–874 (2006)
14. Attia, J.: Moving beyond sensitivity and specificity: using likelihood ratios to help interpret diagnostic tests. Australian Prescriber 26(5), 111–113 (2003)
15. Otsu, N.: Threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. 9(1), 62–66 (1979)

An Integrated System of Face Recognition

Hwei-Jen Lin, I-Chun Pai, and Fu-Wen Yang

Department of Computer Science and Information Engineering
Tamkang University, Taipei, Taiwan, ROC
086204@mail.tku.edu.tw, 695410323@s95.tku.edu.tw,
112405@mail.tku.edu.tw

Abstract. Face Recognition is an important topic in the field of pattern recognition. This technology has a variety of applications including entrance guard control, personal service system, criminal verification, and security verification of finance. Our research focuses on the development of a human face recognition system. It is a challenge to correctly identify a human in an image under various possible situations including difference of lighting conditions, change of hairstyles, variation of facial expression, and different aspects of the face.

We have analyzed several existing face recognition techniques and found that each of them is performed well over some specific sets of testing samples but poorly over some other sets. This motivates us to combine different techniques to construct a better face recognition system. First, we propose a new module E-2DPCA applying DCT for image enhancement and 2DPCA for feature extraction. The experimental results show that the recognition accuracy of E-2DPCA is better than all the modules we have analyzed. We choose the best two from those analyzed and compared them with our proposed E-2DPCA module, and found that although the E-2DPCA module outperforms the other two modules, each of the three modules behaves better than others over some specific set of samples. Thus we combine the three modules and apply weighted voting scheme to choose the recognition result from those given by the three modules. Experimental results show that the integrated system can further improve the recognition rate.

Keywords: Face recognition, principle component analysis (PCA), two dimensional principle component analysis (2DPCA), discrete cosine transformation (DCT), weighted voting, spatial domain, frequency domain, genetic algorithms.

1 Introduction

Face recognition is one of the most important methods for personal identification, since its non-contact process can not only avoid false alarms caused by improper contact but also makes it more friendly than other kinds of biometric identification systems. Face recognition has a wide range of applications, including entrance guard control, digital surveillance system, personal service system, criminal verification, and security verification of finance.

Generally speaking, a face image contains too much information useless for recognition. How to extract the meaningful information of faces is the main issue for a face recognition system, since the use of meaningful information can effectively distinguish among different faces images and reduce the dimension of feature representation; that

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 86–93, 2009.

is both the effectiveness and efficiency of recognition can be improved with the use of meaningful information. Many methods for face recognition acquire meaningful information by extracting linearly independent components in face pattern and choose the most important components for feature representation, including principle component analysis (PCA) [1-11], linear discriminant analysis (LDA) [12][13], singular value decomposition (SVD) [14], and kernel-bases system [15].

Over the methods mentioned above, PCA is of the best recognition accuracy. Since Turk and Pentland [1] first used PCA on face recognition in 1991, PCA became widely applied in this field. PCA selects some most important components from the eigenvectors of the covariance matrix of the samples to form a so-called eigenspace which is of much lower dimension than the sample space. Each sample can be represented by its projection on this space to achieve dimension reduction. K. Shigenari et al. [3] effectively applied orthogonal decomposition and virtualization to their proposed face recognition method.

However, using PCA is storage-consuming and time-consuming on solving the eigenvalues and the eigenvectors for a covariance matrix of face samples, which is usually of huge size. Jian Yang et al. [8] proposed a variation of PCA, called Two Dimensional PCA (2DPCA), which use a variant of covariance matrix of smaller size.

We analyzed and compared different existing face recognition methods, including PCA, 2DPCA, ICA, LDA, and DCT, and found that each of them performed well over some specific sets of testing samples but poorly over some other sets. This motivated us to combine different techniques to construct a better face recognition system.

First, we proposed a new face recognition module, called Enhanced-2DPCA (or E-2DPCA) and proved that the recognition accuracy of E-2DPCA is better than each of the techniques we have analyzed. We chose the best two, namely 2DPCA and DCT, from those we analyzed and compared them with our proposed E-2DPCA module, and found that although the E-2DPCA module outperforms the other two modules, each of the three modules behaves better than others over some specific set of samples.

Thus we combine the three modules and apply weighted voting scheme to choose the recognition result from those given by the three modules. The recognition rate is affected by the given weights. To maximize the recognition rate, we apply the Genetic Algorithm to find the optimal weights for the system.

The remainder of this paper is organized as follows. In Section 2, we introduce the two methods, 2DPCA and DCT. A detailed description for our proposed system is given in Section 3. Section 4 shows the experimental results of the proposed method and compares them with some other methods. Finally, we draw our conclusions and provide suggestions for future work in Section 5.

2 2DPCA and DCT

For a given set S of N images $A_1, A_2, \ldots, A_N$, of size $m \times n$, each image A_i is expressed as a t-dimensional vector x_i (or a $1 \times t$ column matrix), where $t = mn$. Then the covariance matrix is of size $t \times t$, as given in (1), where $\overline{x}$ is the mean image vector. Now evaluate the t eigenvectors of the matrix C and select d ($d << t$) most important ones, say $u_1, u_2, \ldots, u_d$, to span an eigenspace, and project each zero-mean translated image $x - \overline{x}$ onto the space to yield a projected vector y, as shown in (2), where the matrix $U = [u_1, u_2, \ldots, u_d]$ is formed by the selected eigenvectors.

$$C = \sum_{k=1}^{N} (x_k - \overline{x})^T (x_k - \overline{x}) . \tag{1}$$

$$y = U^T (x - \overline{x}) . \tag{2}$$

The number of selected eigenvectors can be determined according to the sum of their corresponding eigenvalues, as shown in (3), where λ_i is the eigenvalue corresponding to the eigenvector u_i, λ_1, λ_2,..., λ_d are the d largest eigenvalues, and θ is a given threshold.

$$(\sum_{i=1}^{d} \lambda_i / \sum_{i=1}^{n} \lambda_i) \geq \theta . \tag{3}$$

However, the covariance matrix of face samples is usually of huge size, and so solving its eigenvalues and eigenvectors is very storage-consuming and time-consuming. J. Yang et al. [8] then proposed a variation of PCA, called Two Dimensional PCA (2DPCA), which varies the definition of the covariance matrix G, as shown in (4), by directly evaluating the covariance of the image matrices without converting them into the vector forms, where $\overline{A}$ is the mean image. Thus, the new covariance matrix is of size n × n, much smaller than the covariance matrix defined by PCA. Now evaluate the n eigenvectors of the matrix G and select f ($f << n$) most important ones, say v_1, v_2,..., v_f, to span an eigenspace, and project each zero-mean translated image $A - \overline{A}$ onto the space to yield a projected vector Y, as shown in (5), where the matrix $V = [v_1, v_2,..., v_f]$ is formed by the selected eigenvectors.

$$G = \frac{1}{N} \sum_{k=1}^{N} (A_k - \overline{A})^T (A_k - \overline{A}) \cdot \tag{4}$$

$$Y = (A - \overline{A}) \cdot X . \tag{5}$$

DCT coefficients have been regarded as good feature for pattern recognition, since most important information can be reserved by just a few of those coefficients. The transformation of images between the spatial domain and the frequency domain of Discrete Cosine is given in the following. For an m × n image $f(x, y)$, let $c(u, v)$ be its DCT coefficients. The DCT is described in (6).

$$c(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f(x,y) \cdot \cos\left[\frac{(2x+1)u\pi}{2m}\right] \cos\left[\frac{(2y+1)u\pi}{2n}\right] \cdot \tag{6}$$

$$\text{where } \alpha(u) = \begin{cases} \sqrt{\frac{1}{2}} & \text{if } u = 0 \\ 1 & \text{otherwise} \end{cases} , \quad \text{for } u \in [0, m-1]$$

$$\text{and } \alpha(v) = \begin{cases} \sqrt{\frac{1}{2}} & \text{if } v = 0 \\ 1 & \text{otherwise} \end{cases} , \quad \text{for } v \in [0, n-1].$$

3 The Proposed System

In this section, we introduce our proposed face recognition module, called Enhanced-2DPCA (E-2DPCA), and then describe how to combine it with two other modules by using a weighted voting scheme to integrate a face recognition system to achieve better performance.

3.1 The E-2DPCA Module

The proposed E-2DPCA module is similar to the 2DPCA module except that both the training images and the testing images are enhanced before being processed. To enhance a given image *A*, we first divide it into non-overlapping blocks of size 8 × 8, perform DCT to each of the blocks, from which the first *r* coefficients along the zigzag order are extracted and transformed back to the spatial domain by using the inverse DCT (IDCT) to obtain a new image *B*. Finally, as described in (7), we take the weighted average of images *A* and *B* to form an enhanced image *C* where $0 < \alpha < 1$.

$$C(\mathrm{x},\mathrm{y}) = (1-\alpha)A(\mathrm{x},\mathrm{y}) + (\alpha)B(\mathrm{x},\mathrm{y}) \tag{7}$$

3.2 Integrating System with a Weighted Voting Scheme

To construct a face recognition system with better recognition accuracy, we combine our proposed E-2DPCA module with the DCT module and the 2DPCA module, and adopting a weighted voting scheme to re-rank the recognition results given by the three modules.

The DCT module first divides each incoming image into non-overlapping blocks of size 8 × 8, performs DCT to each of the blocks, from which the first *r* coefficients along the zigzag order are extracted to be the feature for recognition. The 2DPCA module, as described in the previous section, projects each incoming image onto the pre-trained eigenspace to obtain the feature (the projection of the image) for recognition.

For recognition, the three modules separately recognize a given test image, and each exhibits and ranks the first *k* images it promises from the database. Up to 3*k* images are given (voted) by the three modules, each of which is also assigned a weight by the corresponding module. An image might be voted by more than one module and assigned more than one weights. The sum of weights assigned to each voted image is calculated and the system exhibits and re-ranks these images according to the total weights assigned to them. The performance of weighted voting scheme depends on the given weights. To obtain the optimal weight values, we perform the Genetic Algorithm (GA) as described in the following subsection.

3.3 Weight Optimization by Genetic Algorithm

Let w_{ij} be the weight assigned to the image that is exhibited by module *i* as rank *j*, where i = 1, 2, 3 and j = 1, 2, …, *k*, and so there are 24*k* weights need to be determined. An 8-bit substring is used to encode each weight w_{ij}, of which the first and the last 4 bits represent the integral and decimal parts, respectively, and so $0 \le w_{ij} < 16$.

Thus, it needs 24*k* bits to encode a set of 3*k* weights. The learning process starts with a randomly generated population of individuals (a set of weights) and produces the subsequent populations by means of *reproduction, crossover*, and *mutation* operators [16]. The individuals having the better *fitness* have more chances to be reproduced. We evaluate the fitness by considering the accuracy rate of recognition for the weighted voting scheme.

4 Experimental Results

The experiments were performed on the (corpus of) Yale database [17] which contains 165 images of 15 individuals, as shown in Fig. 1. There are 11 different images for each individual under various facial expressions and lighting conditions. Fig. 2 shows 11 images for one of the 15 persons. Each image was cropped and resized to 80 × 64 pixels.

In our experiments, each of the 11 classes of images was trained. For the enhanced part of the E-2DPCA module, we empirically chose $r = 20$ and $\alpha = 0.5$ for the number of DCT coefficients selected from each those for each block of size 8 × 8, and for the weight used informing an enhanced image, respectively. In the part of feature extraction, we only choose the first 20 ($d = 20$) eigenvectors ($\lambda_1, \lambda_2, \dots, \lambda_{20}$) from the 64 eigenvectors, which can achieve the ratio $(\sum_{i=1}^{20} \lambda_i / \sum_{i=1}^{64} \lambda_i) \cong 0.966$. As shown in Table 2, the proposed E-2DPCA module is on average better than the other two modules.

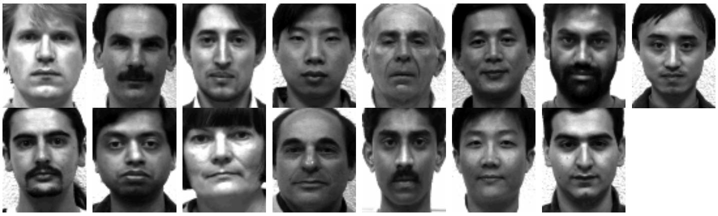
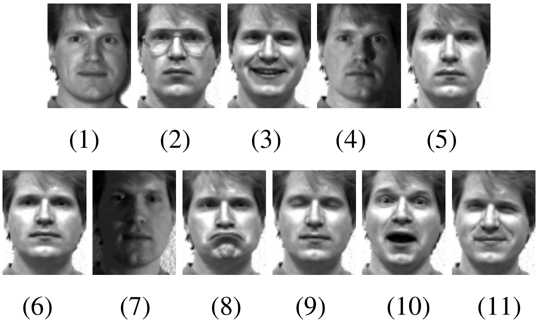


Fig. 1. Examples of Yale database



(1) (2) (3) (4) (5)

(6) (7) (8) (9) (10) (11)

Fig. 2. Examples of 11 different facial expressions and lighting conditions in Yale database

The parameters assigned to the Genetic Algorithm for optimizing the weight set is as follows: population size is 50, crossover probability p_c is 0.5, and the mutation probability p_m is 0.01. The training process stops after 400 generations. The resulting optimal weights are shown in Table 1. With these weights, the integrated system achieves an average recognition rate of 77.96%, and outperforms each of the single modules, as shown in Table 2. Further, as can be found in Fig. 3, the recognition rate of the integrated system with the use of these optimal weights is as high as 89.09% under the training of dataset 9.

Table 1. The optimal weights generated by GAs

methods rank	E-2DPCA	2DPCA	DCT
1	13	8	11.5
2	6.625	0.875	2
3	2.75	0.375	0.875
4	1.0625	0.8125	0.3125

Table 2. Comparison of three modules and the integrated system

modules	E-2DPCA	2DPCA	DCT	Font size and style
recognition rate	74.38%	73.50%	72.73%	77.96%

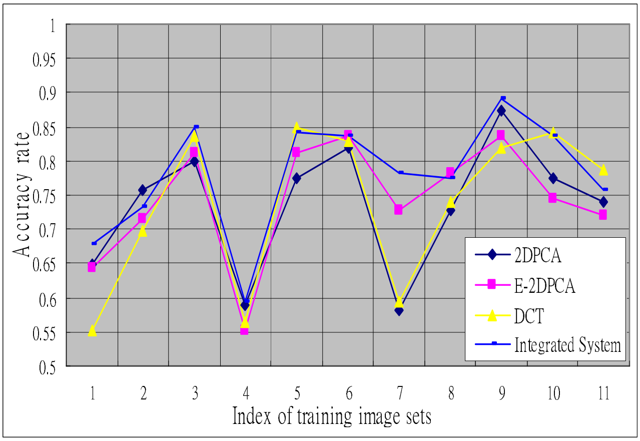


Fig. 3. Comparison of the integrated system with the three single modules over different training datasets

5 Conclusions and Future Research

In this paper, we analyze some face recognition modules and propose a new module E-2DPCA applying DCT for image enhancement and 2DPCA for feature extraction. The experimental results show that the recognition rate of E-2DPCA is on average better than all the modules we analyzed. We choose the best two from those analyzed to combine with our E-2DPCA module and adopt a weighted voting scheme to integrate a better system. The weights used in the weighted voting scheme were obtained and optimized by a genetic algorithm. It is shown by our experiments that the integrated system further improves the recognition rate.

Although the performance with the 7th training set was significantly improved by the integrated system, that with the 4th one was not improved a lot. It is due to the poor lighting condition. The search for an appropriate preprocessing for this kind of images is part of our future research. Besides we would also like to search for better features and a more effective method for face recognition, to further improve the recognition rate for our system.

References

1. Turk, M.A., Pentland, A.P.: Face Recognition Using Eigenfaces. Comput. Vis. Pattern Recognit., 586–591 (1991)
2. Yampri, P., Pintavirooj, C., Daochai, S., Teartulakarn, S.: White Blood Cell Classification based on the Combination of Eigen Cell and Parametric Feature Detection. Papers of Technical Meeting on Medical and Biological Engineering, IEE Jpn. 6, 1–4 (2006)
3. Shigenari, K., Sakaue, F., Shakunaga, T.: Decomposition and Virtualization of Eigenface for Face Recognition under Various Lighting Conditions. Syst. Comput. Jpn. 36(1), 25–34 (2005)
4. Mellakh, M.A., Petrovska-Delacretaz, D., Dorizzi, B.: Using Signal/Residual Information of Eigenfaces for PCA Face Space Dimensionality Characteristics. In: 18th International Conference on Pattern Recognition, vol. 4, pp. 574–577 (2006)
5. Lin, K.H., Lam, K.M., Siu, W.C.: Spatially Eigen-Weighted Hausdorff Distances for Human Face Recognition. Pattern Recognit. 36(8), 1827–1834 (2003)
6. He, J.Z., Zhu, Q.H., Du, M.H.: Face Recognition Using PCA on Enhanced Image for Single Training Images. In: 5th International Conference on Machine Learning and Cybernetics, pp. 3218–3221 (2006)
7. Hong, X., Yao, H., Chen, R.: A PCA Based Visiual DCT Feature Extraction Method for Lip-Reading. In: 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 321–326 (2006)
8. Yang, J., Zhang, D., Frangi, A.F., Yang, J.Y.: Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition. IEEE Trans. Pattern Anal. Mach. Intell. 26, 131–137 (2004)
9. Wen, Y., Shi, P.: Image PCA: A New Approach for Face Recognition. In: International Conference on Acoustics, Speech, and Signal Processing, vol. 1, pp. 1-1241-1-1244 (2007)
10. Sun, W., Ruan, Q.: Two-Dimension PCA for Facial Expression Recognition. In: 8th International Conference on Signal Processing, vol. 3 (2007)
11. Zhang, D., Zhou, Z.H.: (2D)2PCA: 2-Directional 2-Dimensional PCA for Efficient Face Representation and Recognition. Neurocomputing 69, 224–231 (2005)
12. Fisher, R.A.: The Statistical Utilization of Multiple Measurements. Ann. Eugen. 8, 376–386 (1938)
13. MartõÂnez, A.M., Kak, A.C.: PCA versus LDA. IEEE Trans. Pattern Anal. Mach. Intell. 23, 228–233 (2001)

14. He, J.Z., Du, M.H., Pei, S.W., Wan, Q.: Face Recognition Based on Spectroface and Uniform Eigen-Space SVD for one Training Image per Person. In: 5th International Conference on Machine Learning and Cybernetics, vol. 8, pp. 4842–4845 (2005)
15. Li, J.B., Chu, S.C., Pan, J.S., Ho, J.H.: Adaptive Data-Dependent Matrix Norm Based Gaussian Kernel for Facial Feature Extraction. Int. J. Innov. Comput. Inf. Control 3(5), 1263–1272 (2007)
16. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Professional, Boston (1989)
17. Yale database, `http://cvc.yale.edu/projects/yalefaces/yalefaces.html`

Block LDA and Gradient Image for Face Recognition

Chuan-Yu Chang and Ching-Yu Hsieh

Institute of Computer Science and Information Engineering,
National Yunlin University of Science & Technology
123, University Road, Section 3, Douliou, Yunlin 64002, Taiwan
chuanyu@yuntech.edu.tw

Abstract. Face recognition is an important issue in pattern recognition. Linear discriminant analysis (LDA) has been widely used in face recognition. However, the LDA-based face recognition methods usually encountered the small sample size (SSS) problem. The SSS problem occurs when the number of samples is far smaller than the dimensionality of the sample space. Therefore, this paper proposed a modified LDA (called block LDA) to divide the input image into several non-overlapping subimages of the same size, in order to increase the quantity of samples and reduce the dimensions of the sample space. In addition, to reduce the influence of illumination variations, face images were transferred to gradient image. Experimental results show that the proposed method indeed solves the SSS problem with a good recognition rate.

Keywords: Face recognition, linear discriminant analysis, small sample size problem.

1 Introduction

In recent years, face recognition is critical in the film processing, identification card recognition, security systems, and criminal identification systems. Challenges in face recognition include illumination, pose, facial expression, and aging, hair, and glasses.

Many face recognition algorithms have been proposed to solve these problems in the past few decades. Template matching is to detect the positions of specific geometric features on the face (such as the eyes, nose and lips), so that their relative positions can be used as facial features for matching or recognition [1], [2], [3]. The principle component analysis (PCA) is to make the high-dimensional space into a lower-dimensional eigenspace through a proper transformation or projection, while retaining most of the principle components [4], [5]. Independent component analysis (ICA) is the method finding the independent components (ICs) to estimate independent sources only through the information for random variable observed, but without the mixing mechanism and sources [6]. Linear discriminant analysis (LDA), which distances the centroids of different classes and narrows the scatters of the same classes, is used to increase the recognition rate [4], [7], [8], [9].

Among these face recognition methods; PCA and LDA are the most popular ones. However, these methods usually encounter the small size sample (SSS) problem, where the dimensions of image space are more than the sample sizes, making the

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 94–102, 2009.

sample less representative. In addition, the within-class scatter matrix may be singular in LDA, which resulted in fail to calculate the inverse matrix and the transformation matrix.

This paper proposed a modified LDA (called block LDA) to solve the SSS problem, we divided the face image into several non-overlapping subimages of the same size. Accordingly, the number of samples is increased, and the dimension of the sample is decreased. In addition, to reduce the influence of illumination variation, all face images are transformed to gradient images.

Figure 1 shows the flowchart of the proposed method. First, original face image is transformed to gradient image. After that, the gradient image is divided into N^2 subimages. Then, these subimages are projected to feature vectors through LDA. Finally, a Euclidean measurement is applied to determine the recognition result.

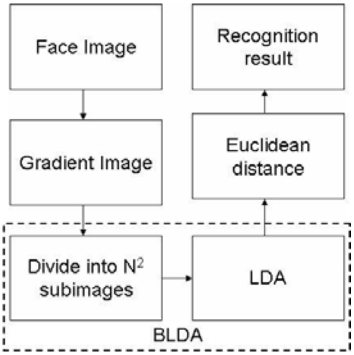


Fig. 1. Flowchart of the proposed method

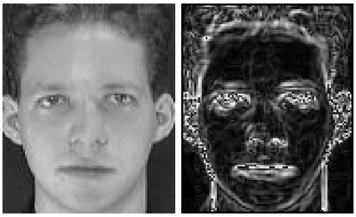
The remainder of this paper is organized as follows: Section 2 describes the gradient image; Section 3 describes traditional LDA and BLDA; Section 4 describes the experimental result; and Section 5 gives the conclusions.

2 Gradient Image

In general, illumination variations should not highly affect the recognition results of human eyes. The dominant factors of face recognition are facial shape and facial features. Thus, this paper transforms the original facial image to gradient image for face recognition. Gradient image contains relative changes in gray level [10], which is less sensitive to illumination variations. Figure 2 shows an original facial image and its corresponding gradient image.

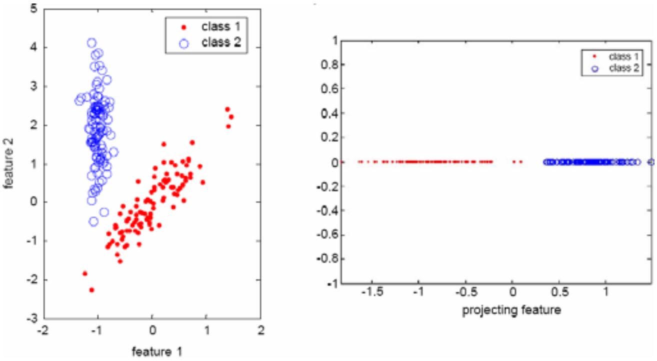
3 Linear Discriminant Analysis

The LDA is able to project an image from a high-dimensional space to a low- dimensional space, thus maximizing the ratio between between-class scatter matrix and within-class scatter matrix. The LDA not only reduces the dimensions, but also distances the distribution of the images. Figure 3 (a) shows synthetic two-dimensional data points belong to two classes. The data along with its one-dimensional LDA transformation are shown in Fig.3 (b). Notice that, the dimensionality reduction and the important discriminatory properties are fully retained.



(a) (b)

Fig. 2. Original image and its corresponding gradient image



(a) (b)

Fig. 3. (a) Original data set distribution. (b) Feature vectors distribution in (a) through LDA.

The detail of the traditional LDA and the proposed BLDA are described as follows.

3.1 Traditional LDA

Assumed that there are C classes of images in the database, each class has M images. The feature vector for the m-th image from the c-th category in the original image is set to $\mathbf{X}_m^c$. The distance between the feature vectors and the centroid of this class, can be expressed by the formula for within-class scatter matrix, $\mathbf{S}_\mathrm{W}$:

$$\mathbf{S}_\mathrm{W} = \sum_{c=1}^{C}\sum_{m=1}^{M}\left(\mathbf{X}_m^c - \overline{\mathbf{X}}^c\right)\left(\mathbf{X}_m^c - \overline{\mathbf{X}}^c\right)^T \tag{1}$$

where $\overline{\mathbf{X}}^c$ is the average of all the feature vectors in the Class c.

The scatter degree among different classes, that is, the distance between the centroids of different classes, can be expressed by the formula for between-class scatter matrix, $\mathbf{S_B}$:

$$\mathbf{S}_{\mathbf{B}} = \sum_{c=1}^{C} \left(\overline{\mathbf{X}}^{c} - \overline{\mathbf{X}}\right)\left(\overline{\mathbf{X}}^{c} - \overline{\mathbf{X}}\right)^{T} \tag{2}$$

where $\overline{\mathbf{X}}$ is the average of all the original feature vectors.

If the within-class scatter matrix is nonsingular, then the best projection matrix $\mathbf{W_{opt}}$ can be determined by the ratio of determinants between between-class scatter matrix and within-class scatter matrix:

$$\begin{aligned} \mathbf{W}_{\mathbf{opt}} &= \arg\max_{W} trace\left\{\frac{\mathbf{W}^{\mathbf{T}}\mathbf{S}_{\mathbf{B}}\mathbf{W}}{\mathbf{W}^{\mathbf{T}}\mathbf{S}_{\mathbf{W}}\mathbf{W}}\right\} \\ &= \arg\max_{W} trace\left\{\mathbf{S}_{\mathbf{W}}^{-1}\mathbf{S}_{\mathbf{B}}\right\} \end{aligned} \tag{3}$$

To find the $\mathbf{W_{opt}}$ which maximizes $\mathbf{S}_{\mathbf{W}}^{-1}\mathbf{S}_{\mathbf{B}}$, the result is derived as $\{w_i|i=1,2,\ldots,p\}$, which is the largest eigenvalue in the first p of $\mathbf{S}_{\mathbf{W}}^{-1}\mathbf{S}_{\mathbf{B}}$, $\{\lambda_i|i=1,2,\ldots,p\}$ is the corresponding eigenvector. They meet the equation below:

$$\mathbf{S}_{\mathbf{W}}^{-1}\mathbf{S}_{\mathbf{B}}\mathbf{w}_{\mathbf{opt}} = \lambda_{\mathbf{i}}\mathbf{w}_{\mathbf{opt}} \tag{4}$$

Basically, only the eigenvectors corresponding to the first p eigenvalues are selected. In this paper, p is selected by taking the first p largest eigenvalues whose total sum accounts for 95% of the sum of the general features. Generally, most energy of images can be kept in this way. The rest of eigenvectors corresponding to the trivial eigenvalues can be ignored.

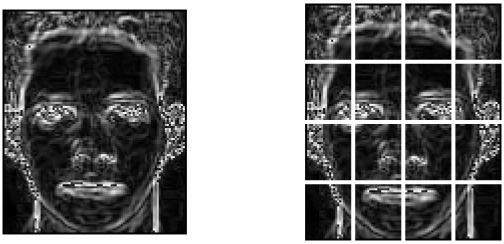
3.2 Block LDA

Since the traditional LDA inheres the SSS problem, that is, when the dimensions of image space is larger than the number of training samples, the within-class scatter matrix becomes a singular matrix, and is unable to calculate its inverse matrix. Thus, the transformation matrix $\mathbf{W_{opt}}$ in Eq.(3) cannot be calculated. Therefore, to solve the SSS problem, this paper proposed a method to divide the input images into several non-overlapping subimages of the same size. It could not only increase the sample size, but also reduce the dimension of the training samples.

In this paper, each input image is divided into N^2 subimages. If the original dimension of the image is $W \times H$, where W is the width of an image and H is the height of an image, and then it is divided into subimages of the size of $W/N \times H/N$. The subimage (j,k) is obtained by:

$$X_{j,k}(\alpha,\beta) = X\left(\frac{W}{N}(j-1)+\alpha, \frac{H}{N}(k-1)+\beta\right) \tag{5}$$

where j,k are between 1 and N; α is between 0 and (W/N)-1; and β is between 0 and (H/N)-1. Figure 4 shows the schematic diagram of image division.



(a) (b)

Fig. 4. The schematic diagram of image division, (a) a gradient image, (b) the gradient image is dividing into 16 subimages

After image division, the within-class scatter matrix $\mathbf{S_W}$ is modified as

$$S_W = \sum_{c=1}^{C}\sum_{m=1}^{M}\sum_{n=1}^{N^2}\left(X_{c,m}^{n} - \overline{X}_c\right)\left(X_{c,m}^{n} - \overline{X}_c\right)^T , \tag{6}$$

where $\overline{X}_c$ represents the average image of all subimages of the c-th class, and is defined as:

$$\overline{X}_c = \frac{1}{MN^2}\sum_{m=1}^{M}\sum_{n=1}^{N^2} X_{c,m}^{n} . \tag{7}$$

The new between-class scatter matrix $\mathbf{S_B}$ is rewritten as

$$S_B = \sum_{c=1}^{C}\left(\overline{X}_c - \overline{X}\right)\left(\overline{X}_c - \overline{X}\right)^T . \tag{8}$$

The average images $\overline{X}$ of all subimages is defined as

$$\overline{X} = \frac{1}{CMN^2}\sum_{c=1}^{C}\sum_{m=1}^{M}\sum_{n=1}^{N^2} X_{c,m}^{n} . \tag{9}$$

Similarity, the optimum projection matrix $\mathbf{W_{opt}}$ is calculated by

$$\begin{aligned}\mathbf{W_{opt}} &= \arg\max_{W} trace\left\{\frac{\mathbf{W^T S_B W}}{\mathbf{W^T S_W W}}\right\} \\ &= \arg\max_{W} trace\left\{\mathbf{S_W^{-1} S_B}\right\}\end{aligned} \tag{10}$$

and satisfies the equation below:

$$\mathbf{S_W^{-1} S_B w_{opt}} = \boldsymbol{\lambda}_\mathbf{i} \mathbf{w_{opt}} \tag{11}$$

As the traditional LDA, the first p largest eigenvalues are obtained. Since some eigenvalues and eigenvectors may be complex numbers, $\|Z\|_2$ is calculated, where $Z=v+oi$, v is the real number part and o is the imaginary number part.

Accordingly, the feature vectors of the n-th subimage is obtained by

$$\boldsymbol{y}^n = \boldsymbol{W}_{opt}{}^T \boldsymbol{X}^n \tag{12}$$

Finally, the feature vectors are combined according to the dividing order as

$$\boldsymbol{Y} = \left\lfloor \boldsymbol{y}^1 \boldsymbol{y}^2 \ldots \boldsymbol{y}^n \right\rfloor \tag{13}$$

During testing, the optimal projection matrix $\mathbf{W_{opt}}$ calculated above is retained. The testing image is also divided into N^2 subimages, and every subimage $\mathbf{X_{testn}}$ is projected by the optimal projection matrix $\mathbf{W_{opt}}$ individually. Thus, the n-th feature vector $\boldsymbol{y}_{test}^n$ of the testing subimages is obtained by:

$$\boldsymbol{y}_{test}^n = \boldsymbol{W}_{opt}^T \boldsymbol{X}_{test}^n \tag{14}$$

Then combine the feature vectors according to the dividing order:

$$\boldsymbol{Y}_{test} = \left\lfloor \boldsymbol{y}_{test}^1 \boldsymbol{y}_{test}^2 \cdots \boldsymbol{y}_{test}^n \right\rfloor \tag{15}$$

The testing image and all the training images are compared to find the one with the shortest Euclidean distance, which is the image with the highest similarity. Assume that there are C classes (people) and each one has m facial images in the training images, the feature vector of the mth image in the c class is $\mathbf{Y}_m^c$, and the feature vector of testing face image is $\mathbf{Y_{test}}$, then the category of the testing image is determined by:

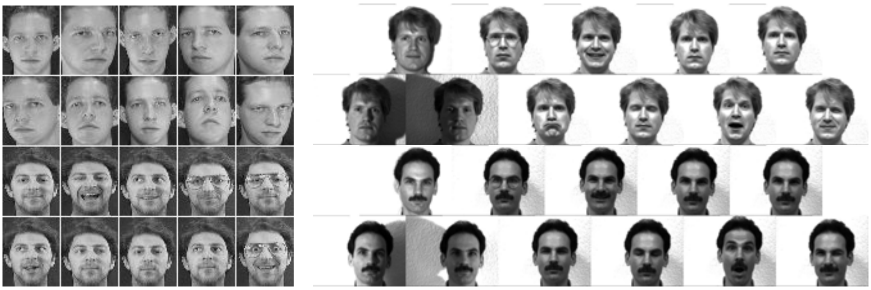
$$\tau = \arg\min_{c} \left\| \mathbf{Y}_{test} - \mathbf{Y}_m^c \right\|_2 . \tag{16}$$

4 Experimental Results

In order to evaluate the performance of the proposed method, two experiments were performed: (1) determination of the number of dividing subimages, (2) the recognition rate comparison with other method and different database.

ORL (Olivetti Research Library) face image database, whose data were collected by AT&T Laboratories Cambridge from April 1992 to April 1994. The database contains 40 people; each has 10 images, which are taken at different times, different facial angles (left, right, look down and up), the different expressions (laugh or not laugh), wear glasses or without glasses. Each image is 92×112, 256 gray-level image with black ground. Samples from the ORL face database are shown in Fig. 5(a).

Yale Database contains 15 people; each one has 11 images with different illumination and facial expressions, and with or without glasses. Each image is 320×243, 256 gray-level image. Samples from the Yale face database are shown in Fig. 5(b).



(a) (b)

Fig. 5. (a) Partial face images from ORL database. (b) Partial face images from Yale database.

A personal computer with a Pentium 4-3.0GHz and 1GB memory was used to perform the experiments. The operating system was Microsoft Windows XP SP2.
The recognition rate is defined as

$$\eta = \frac{R}{A} \times 100\% \tag{17}$$

where R is the number of image correctly recognized and A is the total number of the testing images.

Experiment I: Determination of the number of dividing subimages
Assume that there are C classes (people) in the training image set, in which each class consists M images. To obtain the optimal transform matrix of LDA, the $\mathbf{S_W}$ must be a nonsingular matrix. That is the dimension of a subimage must far bigger than the number of training images.

To solve the SSS problem, each image was divided into $N \times N$ subimages. Then the size of each subimage is $W/N \times H/N$, where W and H are the width and height of an image, respectively.

To determine an appropriate dividing number, various dividing numbers with different databases were performed. For each person, we selected half of the face images (5) as the training set, and the other half (5) as the testing set. Thus, there are 200 training samples and 200 testing samples selected from the ORL database. 75 training samples and 75 testing samples selected from Yale database. The recognition rates under different dividing numbers are shown in Table 1.

The experimental results show that the $\mathbf{S_W}$ is a nonsingular matrix when $N \geq 2$ and $N \geq 6$ in the ORL and in Yale database, respectively.

From Table 1, we obtained the highest recognition rate of 96.5% in ORL database, when the gradient image is divided into 8x8 or 10x10 subimages. In Yale database, we achieved the highest recognition rate of 96% when the dividing number is 18x18 and 20x20.

Table 1. The recognition rates under different dividing numbers

ORL				Yale			
Number of dividing	Number of training sample	Size of subimage	Accuracy rate	Number of dividing	Number of training sample	Size of subimage	Accuracy rate
2×2	800	2576	46%	4×4	1200	4800	singular
3×3	1800	1110	85.5%	6×6	2700	2120	85.33%
4×4	3200	644	90%	8×8	4800	1200	89.33%
5×5	5000	396	96%	10×10	7500	768	93.33%
6×6	7200	270	95.5%	12×12	10800	520	89.33%
7×7	9800	208	95.5%	14×14	14700	374	94.66%
8×8	12800	154	96.5%	16×16	19200	300	93.33%
9×9	16200	120	96%	18×18	24300	221	96%
10×10	20000	99	96.5%	20×20	30000	192	96%

Experiment II: Face recognition comparison with different methods and different databases

To show the proposed method has capability of face recognition with good performance. PCA, LDA, Fuzzy fisherface[8], ESTM[11], and the proposed method were compared. The recognition rates are summarized in Table 2.

According to Table 2, the proposed method achieved the highest recognition rate of 96.5% in the ORL and 96% in the Yale databases, respectively. The ESTM[11] method uses template-based classification. The recognition rate of this template-based method is 78.3% for the ORL database and 88.7% for the Yale database. Because template-based approaches are based on similarity measurement of two feature sets without consideration of invariant salient feature, the recognition results highly depended on the variations of scale, pose, and shape. Even a slight variation in head pose significantly affected the similarity measurements. The PCA, LDA, and Fuzzy fisher face methods are statistics-based approach that had a higher recognition rate in situations with different head poses and facial expressions (ORL database). However, the statistic-based method is sensitive to illumination variance (Yale database). Among the statistics-based methods, the Fuzzy fisherface obtained the highest recognition result of 95% in ORL database and 94.8% in Yale database, respectively.

Table 2. Accuracy of face recognition with different methods

Database Method	Accuracy rate	
	ORL	Yale
PCA	76%	79.6%
LDA	80%	91%
Fuzzy fisherface[8]	95%	94.8%
ESTM[11]	78.3%	88.7%
Proposed method	96.5%	96%

5 Conclusions

In general, variation in illumination seriously affects the recognition results; hence in this paper, the original facial images were transformed to gradient images to reduce the sensitivity of illumination variation.

To solve the SSS problem of the traditional LDA, a modified LDA (block LDA) is proposed. The gradient images were dividing into several subimages of the same size without overlapping to increase the quantity of training samples, and reduce the dimensions of training sample.

Experimental results show that the proposed method substantially obtained a higher recognition rate than traditional LDA and other methods. In addition, the proposed method can attenuate the influences of illumination, facial expression, and pose variations.

Acknowledgement. This work was supported by the National Science Council, Taiwan, under grant NSC 96-2218-E- 224-007.

References

1. Bae, H., Kim, S.: Real-time face detection and recognition using hybrid-information extracted from face space and facial features. Image and Vision Computing 23, 1181–1191 (2005)
2. Bruneli, R., Poggio, T.: Face recognition: feature versus templates. IEEE Trans. on Pattern Analysis and Machine Intelligence 15, 1042–1052 (1993)
3. Arca, S., Campadelli, P., Lanzarotti, R.: A face recognition system based on automatically determined facial fiducial points. Pattern Recognition 39, 432–443 (2006)
4. Gottumukkal, R., Asari, V.K.: An improved face recognition technique based on modular PCA approach. Pattern Recognition Letter 25, 429–436 (2004)
5. Zhao, Z.Q., Huang, D.S., Sun, B.Y.: Human face recognition based on multi-features using neural networks committee. Pattern Recognition Letter 25, 1351–1358 (2004)
6. Barthlett, M.S., Movellan, J.R., Sejnowski, T.J.: Face recognition by independent component analysis. IEEE Transaction on Neural Networks 13, 1450–1464 (2002)
7. Zhou, D., Yang, X., Peng, N., Wang, Y.: Improved-LDA based face recognition using both facial global and local information. Pattern Recognition Letter 27, 536–543 (2006)
8. Kwaka, K.C., Pedryczb, W.: Face recognition using a fuzzy fisherface classifier. Pattern Recognition 38, 1717–1732 (2005)
9. Zheng, W., Zhao, L., Zou, C.: An efficient algorithm to solve the small sample size problem for LDA. Pattern Recognition 37, 1077–1079 (2004)
10. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Prentice-Hall Internation, Inc., Englewood Cliffs (2002)
11. Xie, X., Lam, K.M.: Elastic shape-texture matching for human face recognition. Pattern Recognition 41, 396–405 (2008)

Ranking Answers by Hierarchical Topic Models

Zengchang Qin[1,⋆], Marcus Thint[2], and Zhiheng Huang[1]

[1] BISC Group, EECS Department, University of California Berkeley, USA
[2] Computational Intelligence Group, Intelligent Systems Lab, BT Group, UK
zcqin@berkeley.edu, zhiheng@cs.berkeley.edu, marcus.2.thint@bt.com

Abstract. Topic models are hierarchical probabilistic models for the statistical analysis of document collections. It assumes that each document comprises a mixture of latent topics and each topic can be represented by a distribution over vocabulary. Dimensionality for a large corpus of unstructured documents can be reduced by modeling with these exchangeable topics. In previous work, we designed a multi-pipe structure for question answering (QA) systems by nesting keyword search, classical Natural Language Processing (NLP) techniques and prototype detections. In this research, we use those technologies to select a set of sentences as candidate answers. We then use topic models to rank these candidate answers by calculating the semantic distances between these sentences and the given query. In our experiments, we found that the new model of using topic models improves the answer ranking so that the better answers can returned for the given query.

1 Introduction

Question answering (QA) is an important area in information retrieval. It involves query analysis, recognition of relevancy and search. In our previous work, we designed a deduction engine which supports a "multi-pipe" process flow to handle keyword search as well as some special prototypes [12]. By reasoning based on these prototypes, our systems can improve over classical keyword matching approach. However, the fundamental problem for learning from text and natural language processing is how to learn the 'meaning' and 'usage' of words in data-driven fashion. How to model polysemy and synonymy of words become the first step towards semantic understanding.

Latent semantic indexing (LSI) [5] a is well-known technique which partially addresses this problem. LSI makes three claims: semantic information can be derived from a word-document co-occurence matrix; that dimensionality reduction is an essential part of this derivation; and the words and documents can be represented as points in Euclidean space. The key idea is to map high-dimensional vocabulary count vectors to a lower dimensional representation by using Singular Value Decomposition and selected largest eigenvalues. Due to the unsatisfactory theoretical foundation, Hofmann developed probabilistic latent semantic

[⋆] Currently at Intelligent Computing and Machine Learning Lab, School of Automation Science and Electrical Engineering, Beihang University, China.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 103–112, 2009.

indexing (pLSI) [9] based on a mixture decomposition derived from a latent class model. pLSI models each word in document as a sample from a mixture of topics. Each word is generated from a single topic, and different words in a document may be generated from different topics. However, pLSI does not provide a probabilistic model at the document level. Blei *et al.* [4] later proposed a three-level hierarchical Bayesian model called latent Dirichlet allocation (LDA). In LDA, document level is modeled by a Dirichlet distribution. LDA has been heavily cited in machine learning community for its effectiveness and theoretical soundness.

Griffiths and Steyvers [7] proposed similar models for learning natural language by using these latent topics, we call these models "topic models". In this paper, we discuss an application using topic models proposed in [13] for answer ranking in a question answering system. This paper is organized as follows: technical details of topic models are introduced in section 2. In the section 3, after a brief introduction on our previous work on question answering system, we discuss how to rank answers by using topic models. Some experimental results on a small corpus are presented in section 4. The conclusions and discussions are given in the final section.

2 Topic Model

The study of latent topics is popularized by Hofmann's work [9] on probabilistic latent semantic indexing. In this model, a document label d ($d = 1, \cdots, D$) and a word w_i ($i = 1 \cdots W$) are conditionally independent given a latent topic z:

$$p(d, w_i) = p(d) \sum_z p(w_i|z)P(z) \tag{1}$$

The model learns the topic mixture $p(z|d)$ only for those training documents and the size of parameters grows linearly with the corpus size M. LDA overcomes these problems by treating the topic mixture weights as a k-dimensional random variable θ with Dirichlet distribution [4].

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \cdots \theta_k^{\alpha_k - 1} \tag{2}$$

Then, the probability of a document $\mathbf{w}$ becomes the conditional probability given hyper-parameters α and γ [4]:

$$p(\mathbf{w}|\alpha, \gamma) = \int p(\theta|\alpha) \prod_{i=1}^{W} \sum_{z_i} p(z_i|\theta) p(w_i|z_i, \gamma) d\theta \tag{3}$$

where γ is a $k \times W$ matrix with $\gamma_{jl} = P(w_i = l|z_i = j)$ and $p(w_i|z_i, \gamma)$ is a multinomial probability conditioned on the topic z_i. More details about LDA can be found in [4].

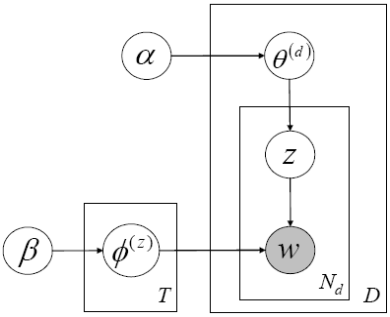


Fig. 1. A general structure of a topic model. The shaded node is the observable variable and others are latent variable. Relationships between latent topic z, random variable θ, ϕ and hyperparameters α and β are discussed in Section 2 below.

Griffiths and Steyvers [7,13] explored a variant of LDA by introducing another multinomial random variable ϕ to smooth the word distribution in every topic. ϕ is also a Dirichlet distribution governed by the hyperparameter β.

$$p(\phi|\beta) = \frac{\Gamma(\sum_{i=1}^{k} \beta_i)}{\prod_{i=1}^{k} \Gamma(\beta_i)} \phi_1^{\beta_1 - 1} \cdots \phi_k^{\beta_k - 1} \tag{4}$$

Figure 1 shows the graphical relations between these variables. The topic model is a generative model [10] for documents. Documents are assumed to be generated following a simple probabilistic procedure. Each document is a mixture of topics and each topic is a probability distribution over words [1,13]. There are set of parameters governed by some prior distributions and these priors are defined by Dirichlet distributions. The variables ϕ, θ and z (the assignment of word tokens to topics) are latent variables and hyperparameters α and β are constants in the model. The inner plate over z and w illustrates the repeated sampling of topics and words until N_d words have been generated for document d. The plate surrounding $\theta^{(d)}$ illustrates the sampling of a distribution over topics for each document d for a total D documents (the whole corpus). The plate for $\phi^{(z)}$ illustrates the sampling of word distributions for each topic z until T topics have been generated [13]. Figure 2 shows the geometric interpretation of the relations between document-topic and topic-word. Suppose we only have 3 words in our vocabulary, then a topic can be represented as a probability distribution on these 3 words. Therefore, a topic must lie on the simplex of these 3 words. Similarly, if we only have 3 topics, a document can be represented as a point on the simplex of topics.

There are a few methods to estimate the parameters for graphical models, such as variational methods [4] and Expectation Maximization (EM) algorithm [14]. In this paper, we use Gibbs sampling algorithm to consider each word token in the text in turn, and estimate the probability of assigning the current word token to each topic, conditioned on the topic assignments to all other word

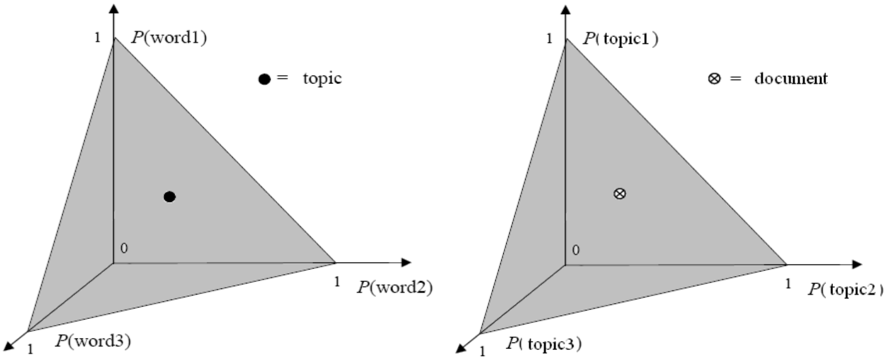


Fig. 2. A topic is a distribution over observable words and a document is a distribution of latent topics. In this simple case, there are 3 words and 3 topics. A document can be represented as a point on the surface of a simplex of topics. A topic can be regarded as a point of on the simplex of words.

tokens. The probability $z_i = j$ (assign token i to topic j) is conditioned on w_i (current word token), $\mathbf{z}_{-i}$ (topic assignments of all other word tokens), and d_i (current document).

$$P(z_i = j|\mathbf{z}_{-i}, w_i, d_i, I) \propto \left(\frac{C^{WT}_{w_i,j} + \beta}{\sum_{w=1}^{W} C^{WT}_{wj} + W\beta} \right) \left(\frac{C^{DT}_{d_i,j} + \alpha}{\sum_{t=1}^{T} C^{DT}_{d_i,t} + T\alpha} \right) \tag{5}$$

where I refers to all other observed information such as all word and document indices $\mathbf{w}_{-i}$ $\mathbf{d}_{-i}$ and hyperparameters α and β. $\mathbf{C}^{WT}$ and $\mathbf{C}^{DT}$ are the matrices of counts with dimensions $W \times T$ and $D \times T$, respectively; C^{WT}_{wj} contains the number of times word w is assigned topic j, not including the current instance i and C^{DT}_{dj} contains the number of times topic j is assigned to some word token in document d, not including the current instance i. The estimate of posterior probability distributions for θ (topic-document distribution) and ϕ (word-topic distribution) directly obtained by [13]:

$$\hat{\phi}^j_i = \frac{C^{WT}_{ij} + \beta}{\sum_{k=1}^{W} C^{WT}_{kj} + W\beta} \tag{6}$$

$$\hat{\theta}^d_j = \frac{C^{DT}_{dj} + \alpha}{\sum_{k=1}^{T} C^{DT}_{dk} + T\alpha} \tag{7}$$

The detailed sampling algorithm and justification of above equations are available at [7,13].

Given a new document $\mathbf{w}$ which is not contained in the training corpus D, how can we represent the document by a distribution of topics? For a particular topic t_j, according to Bayes' rule:

$$p(t_j|\mathbf{w}, D) \propto p(\mathbf{w}|t_j, D)p(t_j|D) \tag{8}$$

where the 'prior' probability of topic t_j (conditional on the training corpus) can be calculated by:

$$p(t_j|D) = \frac{\sum_{i=1}^{W} C_{w_i,j}^{WT}}{\sum_{i=1}^{W} \sum_{j=1}^{T} C_{w_i,j}^{WT}} \tag{9}$$

and the likelihood is:

$$p(\mathbf{w}|t_j, D) = \sum_{d}^{|D|} \prod_{i} p(w_i|t_j)p(t_j|d)p(d) \tag{10}$$

For training the topic model in Figure 1, we have to predefine the number of topics. Automatic determination of the number of clusters has been a persisting challenge in machine learning though some work has tried to address this problem [1]. In this work, we simply decide the topic number based on the size of corpus. For example, we set 20% of the corpus size as the topic number. For a new document, we can then calculated the topic distribution according to equation 8. Some recent work are looking at more general topic models by considering the correlation between topics [3] and the time evolution of topics [2] in large corpus. In this paper, we assume the topics are conditionally independent without evolutionary properties because the corpus we are testing only contains some simple texts from user manuals. In the following section, we discuss the use of this model for ranking answers given a query.

3 Ranking Answers

In our previous work [11,12], we described a hybrid reasoning engine which supports a "multi-pipe" process flow to handle Precisiated Natural Language (PNL) based deduction as well as other natural language phrases that do not match PNL protoforms. The resulting process flows in a nested form, from the inner to the outer layers: (a) PNL-based reasoning where all important concepts are pre-defined by fuzzy sets, (b) deduction-based reasoning which enables responses drawn from generated/new knowledge, and (c) key phrase based search when (a) and (b) are not possible. The design allows for two levels of response accuracy improvement over standard search, while retaining a minimum performance level of standard search capabilities.

In this research, we add the topic model to the end of our pipeline design. We use topic models to measure the 'semantic' distance between each candidate answers and the query. Closer the semantic distance implies closer semantic relation between the answers and the query. This information is used to re-rank the candidate answers selected by other NLP tools.

One challenge in this application is that, we usually have a short list of keywords for the query and each candidate answer sentence. Hence, the topic distribution calculated on these words may not be very accurate when there is some semantic meaning behind the words we cannot capture. For example, in a query "Where can I buy a Ford near Berkeley", a human can understand that

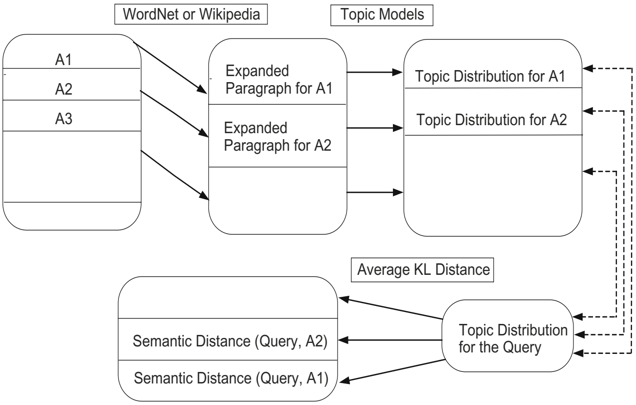

WordNet or Wikipedia
Topic Models
A1
A2
A3
Expanded Paragraph for A1
Expanded Paragraph for A2
Topic Distribution for A1
Topic Distribution for A2
Average KL Distance
Semantic Distance (Query, A2)
Semantic Distance (Query, A1)
Topic Distribution for the Query

Fig. 3. Topic Models for answer ranking: we selected a subset of candidate answers by using keyword search or other NLP techniques. Each keyword in the answer is expanded by WordNet or Wikipedia. We calculate the topic distributions for answers and the query and compare the distance between them. The average Kullback-Leibler divergence is used to measure the semantic distances between candidate answers and the query.

someone is trying to buy a car of brand Ford. Even if there are some sentences about auto dealer in San Francisco Bay Area, this useful information cannot be found by key-word search and other classical NLP tools. We propose a sentence augmentation approach to expand each short sentence (including query) into a paragraph by using WordNet [8] or Wikipedia. We use the descriptions on those keywords (mainly nouns) to formulate a new paragraph which later on will be used to calculate topic distribution. By using Wikipedia, we would select the first two sentences. For example, word ‘Ford’ becomes

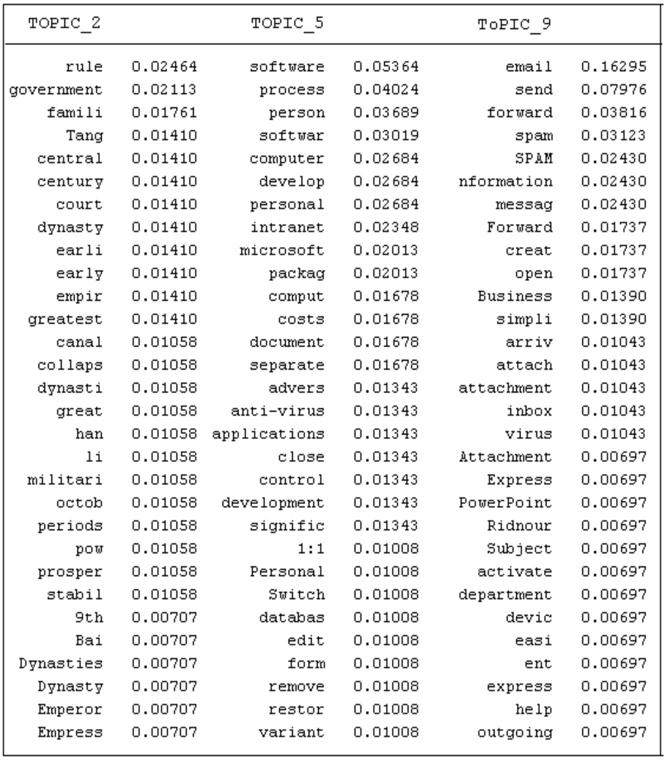
> Ford Motor Company is an American multinational corporation and the world’s third largest **auto** maker based on worldwide vehicle sales. Based in Dearborn, Michigan, a suburb of Detroit, the automaker was founded by Henry Ford and incorporated in June 16, 1903.

and ‘Berkeley’ becomes

> Berkeley is a city on the east shore of **San Francisco Bay** in Northern California, in the United States. Its neighbors to the south are the cities of Oakland and Emeryville.

Although the expansion injects noise to the query and answer sentences, we can see that addition of valuable keywords ‘auto’ and ‘San Francisco Bay’ helps to semantically link ‘Ford’ and ‘Berkeley’, respectively. The basic process is illustrated in Figure 3. The semantic distance between answer A and query Q is measured by the average Kullback-Leibler (AKL) Distance (or divergence):

$$AKL(A||Q) = \frac{KL(A||Q) + KL(Q||A)}{2} \tag{11}$$



TOPIC_2		TOPIC_5		ToPIC_9	
rule	0.02464	software	0.05364	email	0.16295
government	0.02113	process	0.04024	send	0.07976
famili	0.01761	person	0.03689	forward	0.03816
Tang	0.01410	softwar	0.03019	spam	0.03123
central	0.01410	computer	0.02684	SPAM	0.02430
century	0.01410	develop	0.02684	nformation	0.02430
court	0.01410	personal	0.02684	messag	0.02430
dynasty	0.01410	intranet	0.02348	Forward	0.01737
earli	0.01410	microsoft	0.02013	creat	0.01737
early	0.01410	packag	0.02013	open	0.01737
empir	0.01410	comput	0.01678	Business	0.01390
greatest	0.01410	costs	0.01678	simpli	0.01390
canal	0.01058	document	0.01678	arriv	0.01043
collaps	0.01058	separate	0.01678	attach	0.01043
dynasti	0.01058	advers	0.01343	attachment	0.01043
great	0.01058	anti-virus	0.01343	inbox	0.01043
han	0.01058	applications	0.01343	virus	0.01043
li	0.01058	close	0.01343	Attachment	0.00697
militari	0.01058	control	0.01343	Express	0.00697
octob	0.01058	development	0.01343	PowerPoint	0.00697
periods	0.01058	signific	0.01343	Ridnour	0.00697
pow	0.01058	1:1	0.01008	Subject	0.00697
prosper	0.01058	Personal	0.01008	activate	0.00697
stabil	0.01058	Switch	0.01008	department	0.00697
9th	0.00707	databas	0.01008	devic	0.00697
Bai	0.00707	edit	0.01008	easi	0.00697
Dynasties	0.00707	form	0.01008	ent	0.00697
Dynasty	0.00707	remove	0.01008	express	0.00697
Emperor	0.00707	restor	0.01008	help	0.00697
Empress	0.00707	variant	0.01008	outgoing	0.00697

Fig. 4. The first 30 words with their distributions in the samples of topic 2, 5 and 9

The score of a set of candidate answers A_i $(i = 1, \cdots, |\mathbf{A}|)$ is calculated by:

$$S(A_i) = \frac{AKL(A_i||Q)^{-1}}{\sum_{i=1}^{|\mathbf{A}|} AKL(A_i||Q)^{-1}} \tag{12}$$

This normalization will assign higher scores to the answers which are more semantically close the query.

4 Experimental Studies

In order to test the effectiveness of the topic model, we compare the keyword based search engine and the new system of combining keyword search and the topic model. We use Lucene [6] as the standard keyword search engine. We applied the topic model to the results from Lucene and the overall score is unweighted sum of the Lucene score and the topic model score. We tested these two systems on the test corpus which contains 110 documents. Over half of them are about telecommunications, the other half documents are about sports, travel, history and other random documents from the Internet. We proposed 90 questions whose correct answers are available in the corpus.

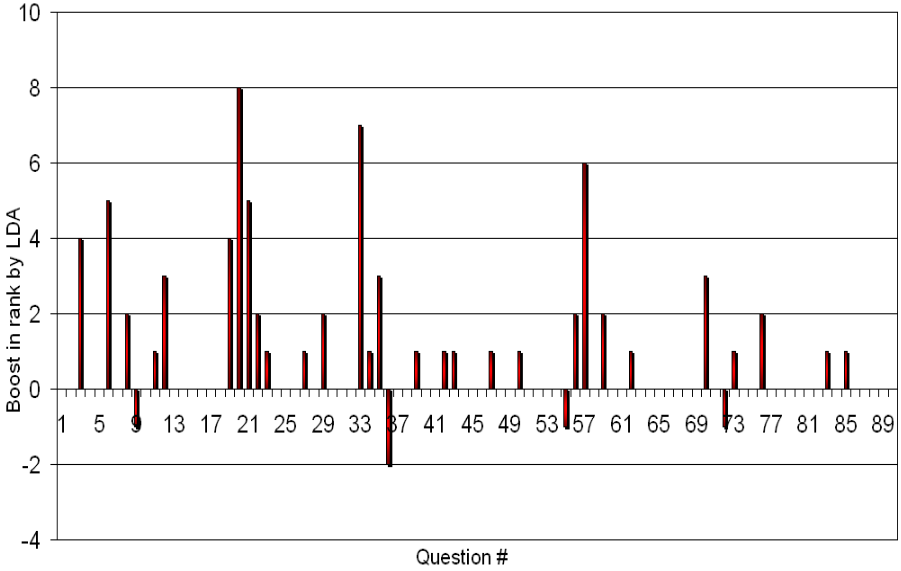


Fig. 5. The rank difference for Lucene and Lucene+LDA on the BT corpus. The horizontal axis is the question number and the vertical axis is the boosted rank of the best answer for that question.

In our offline training, we use 20 topics (about 20% of the corpus size) and the first 30 most frequent words in topic 2, 5 and 9 are shown in Figure 4. For each topic, the numbers after words are the probabilities of these words in this topic. Although there is no explicit name for each topic, we still can see that topic 2 is about history (Tang dynasty of China), topic 5 is about computer and internet, and topic 9 is about spam emails. Since these words occur together more frequently, we may consider that they are semantically close. By using the topic model, we can cluster these words into one semantic topic.

Figure 5 illustrates the results of the two systems (Lucene and Lucene+LDA) on a test BT corpus. Each bar represents the difference between two ranks given by Lucene and Lucene+LDA on the same question. For example, given Question 6, the best answer is ranked as 10th by Lucene and ranked 5th by Lucene+LDA. We obtained the difference 5 by using:

$$D = Rank_{(Lucene)} - Rank_{(Lucene+LDA)} \tag{13}$$

where $Rank_{Lucene}$ is the cardinal number of the best answer rank given by Lucene. Therefore: if the difference is positive, it means that the new system boosts the best answer with high ranks; if negative, it means that the new system actually performs worse ranking than Lucene due to the extra augmentation noise; if zero, it means that there is no significant differences for these two systems. In the 90 questions, the new system improve ranking of the basic keyword matching approach for 29 questions. There are also 4 questions that the new system obtained worse ranking. For the remaining 57 questions, these two systems give the same rankings of the best answers. The bad rankings for questions #9, #50 #83 and #88 are likely due to the noise injected by keyword augmentation. Further research is still needed to reduce such noise in terms of an optimal augmentation strategy.

The following output is an example of better ranking by using topic models. Given the question: **What is the oil price per barrel in October?** The Lucene results give the best answer ranked as number 10: The new system (Lucene+LDA) boosted the same answer to 5th rank. The scores of Lucene and LDA are shown at below:

```
5: doc: QAtestDocs/senDocLucene/QAtestDocs-BT-finance2&5.txt
with lucene: 0.147376507520675 and LDA: 0.23581354260208 (Final
Score: 0.38319005012276147)
Content: Light sweet crude for October delivery rose 32 cents to
settle at $70.03 a barrel on the New York Mercantile Exchange.
```

The benefit of such rank boosting is that when the QA system selects the top N-ranked sentences as candidate answers the best answer is more likely to be included and shown the the final answer list.

5 Conclusions

In this paper, we investigated a methodology for using hierarchical probabilistic models for question answering. Topic models are used to re-rank the candidate answers from a standard keyword based search engine. Each candidate answer and the query is represented by the distributions over latent topics from offline training of topic models. Because there is only a short list of keywords for candidate answer sentences and query, their keywords were expanded by WordNet before the topic distribution calculation.

We compared the new system and the basic keyword search engine on a small test corpus. The ranking results of 90 sample questions are presented. The new systems performs better than the basic keyword matching for about 31% of the questions, for 66% of questions, the new system performs as good as the basic keyword matching. There is also about 3% of questions that the new system performs worse due to the noise of augmentation. By considering the semantic information, topic models can improve the ranking made by classical search engine. It also provides a new research direction of how to efficiently use probabilistic models to improve the answer ranking in question answering.

Acknowledgements

Qin and Huang are funded by the British Telecom (BT)/BISC Research Fellowship.

References

1. Blei, D.M., Griffiths, T., Jordan, M.I., Tenenbaum, J.: Hierarchical Topic Models and the Nested Chinese Restaurant Process. In: Thrun, S., Saul, L., Schoelkopf, B. (eds.) Advances in Neural Information Processing Systems (2004)
2. Blei, D.M., Lafferty, J.D.: Dynamic Topic Model. In: Proceedings of the 23rd ICML, Pittsburgh, USA (2006)
3. Blei, D.M., Lafferty, J.D.: Correlated Topic Models. In: Advances in Neural Information Processing Systems 18. MIT Press, Cambridge (2006)
4. Blei, D.M., Ng, A., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Leanring Research 3, 993–1022 (2003)
5. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the American Society of Information Science 41 (1990)
6. Gospodnetic, O., Hatcher, E.: Lucene in Action, Manning (2004)
7. Griffiths, T.L., Steyvers, M.: Finding Scientific Topics. Proceedings of the National Academy of Science 101, 5228–5235 (2004)
8. Miller, G.: WordNet: a lexical database. Communications of the ACM 38(11), 39–41 (1995)
9. Hofmann, T.: Probabilistic Latent Semantic Anlalysis. In: Proceedings of UAI 1999, Stockholm (1999)
10. Jordan, M.I.: Learning in Graphical Models. MIT Press, Cambridge (1999)
11. Beg, M.M.S., Thint, M., Qin, Z.: PNL-enhanced Restricted Domain Question Answering System. In: The Proceedings of IEEE-FUZZ, London, pp. 1277–1283. IEEE Press, Los Alamitos (1999)
12. Qin, Z., Thint, M., Beg, M.M.S.: Deduction Engine Designs for PNL-based Question Answering Systems. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 253–262. Springer, Heidelberg (2007)
13. Steyvers, M., Griffiths, T.: Probabilistic topic models. In: Landauer, T., McNamara, D., Dennis, S., Kintsch, W. (eds.) Latent Semantic Analysis' A Road to Meaning (2007)
14. Wei, X., Croft, W.B.: LDA-based document models for ad-hoc retrieval. In: SIGIR 2006, Seattle, WA (2006)

Using Genetic Process Mining Technology to Construct a Time-Interval Process Model

Chieh-Yuan Tsai* and I-Ching Chen

Department of Industrial Engineering and Management,
Yuan-Ze University, Chung-Li, Taoyuan, Taiwan
cytsai@saturn.yzu.edu.tw

Abstract. To understand process executed in many activities, process mining technologies are now extensively studied. However, three major problems in the current process mining techniques are identified. First, most process mining techniques mainly use local search strategy to generate process models. Second, time intervals between two actives are not considered so that patterns that are different in view of time are regarded as the same behaviors. Third, no precision evaluation measure is defined to evaluate the quality of process models. To solve these difficulties, this research proposes a time-interval process mining method. A genetic process mining algorithm with time-interval consideration is developed. Then, a precision evaluation measure is defined to evaluate the quality of the generated process models. Finally, the best process model with highest precision value is reported.

Keywords: Process Mining, Genetic Algorithm, Time-interval.

1 Introduction

For the past several years, some famous process mining techniques including α-algorithm [1], α^+-algorithm [2], β-algorithm [3] have been developed. These methods use event logs to discover process models. Although these process mining techniques can help managers realize the sequence and relation of tasks from discovered models, some problems have been revealed. First, theses process mining techniques such as α-algorithm are mainly based on the local strategy which builds the process models step by step based on local information [4, 5]. For this reason, these techniques are hard to deal with noisy logs well. In addition, invisible tasks, duplicate tasks, non-free choice, and loops are also difficult to be handled [6]. Hence, process models built using local information search approaches could be problematic. Another major problem is that most generated process models do not involve the time information such as time-stamp and time-interval. That is, most process models reveal only the information of sequences and flows between tasks instead of how much time taken between tasks. However, the tasks happened between different time-intervals should be considered as different behavior. For example, customers who bought product A then product B within one to two weeks should be regarded as different behavior in which customers

[*] Corresponding author.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 113–122, 2009.

bough product A then product B within three to four months. Therefore, without considering time-intervals between tasks, the process model could be too rough to help managers take the right marketing actions at the right time. The other problem is the lack of a precision evaluation measure to evaluate the quality of process models in existing process mining techniques. A process model with high quality means that this process model can represent the behaviors in the event logs precisely. That is, a high quality process model should be able to predict the future behavior in a correct way. Unfortunately, few studies clearly define how to evaluate the quality of a process model, which is an important issue in the process mining research.

To overcome the drawbacks mentioned above, a GA based process mining algorithm known as genetic process mining [4, 7] is applied to avoid the problems of local search. During conducting the genetic process mining, time-interval information between tasks is considered so that behaviors with different time relationship can be distinguished well. In addition, a precision evaluation measure is defined to evaluate the quality of process models. With the measure, managers can select the best process model among a set of candidate models and make a better decision making.

2 Process Mining

The goal of process mining is to obtain the information about processes from event logs. Process mining is also called *process discovering* while it is discussed in the context of software engineering [8] and *workflow mining* while it is discussed in the business sense [9]. Cook *et al.* is the first ones to mine the model from event log in the context of software engineering [10]. They introduced the RNet, KNet and Markov methods for process discovering. Subsequently, Cook and Wolf extended their study for discovering process model on concurrent process [11]. They used particular metrics: entropy, event type counts, periodicity and causality to discover process models from event stream. Agrawal *et al.* is the first ones to apply the process mining in business filed [9]. Their process models are based on the workflow graph that contains the nodes and edges where the nodes represent the activities and edges represent the relations between activities. They solved two problems for process mining. One is to discover the adaptive workflow graph from the logs. Another is to find the definition of edge condition.

The challenging problems in process mining are mining hidden tasks, mining duplicate tasks, mining non-free-choices constructs, mining loops, the using time, dealing with noise and dealing with incompleteness [12]. The α-algorithm proposed by Aalst *et al.* [1] is a major approach in process mining for discovering the process models (Petri nets) by ordering relations between events (tasks) from the event logs. De Medeiros *et al.* [6] introduced the limitations existed in the α-algorithm and these limitations cannot be used in the α-algorithm. These limitations are short loop, invisible tasks, duplication task, noise, implicit place and non-free choice mentioned above. In order to solve these limitations, an extension approach of α-algorithm, called α^{+}-algorithm is provided by De Medeiros *et al.* [2]. They used α^{+}-algorithm to handle the short loops. Wen *et al.* [13, 14] proposed α^{++}-algorithm to deal with non-free-choice constructs. Furthermore, De Medeiros *et al.* proposed a genetic process mining method to deal with the noise and limitations in current techniques of process mining such as the α-algorithm [4, 7]. They indicated that genetic process mining method can do the better in deal with noise from the event log. Additionally, Ren *et al.* [3] proposed the β-algorithm to deal with the short loops limitation existed in the α-algorithm. In [6, 15], Aalst *et al.* made a survey of process mining and introduced some typical approaches, and process mining tools.

3 Research Method

The framework of this research is shown in Fig. 1(a). The proposed mining approach starts with the time-interval analysis from an event log. To solve the process mining techniques mainly based on the local search strategy, a genetic process mining method with global search strategy is applied to construct the process model. Four important parameters including the population size, the number of generation, crossover rate and mutation rate in genetic process mining method should be determined. Different combinations of these four parameters will construct different process models. To select the best process models, a precision evaluation measure is developed so that the quality of the constructed process models can be calculated. The precision evaluation measure is composed of two precision measurements. The first precision measurement evaluates the constructed process models based on segment sequences, while the second precision measurement evaluates the constructed process model based on whole sequences. The k-fold cross-validation method is applied to derive the precision of each process model. The training dataset is used to construct a process model using the genetic process mining method and the testing dataset is used to evaluate the quality of the constructed process models. Note that, only one genetic parameter is changed in each experiment. Finally, a process model with highest precision will be selected.

3.1 Event Logs and Time-Interval Analysis

An event log records a series of events in which each event (task) is represented by a case identifier, a task identifier, and a timestamp. One event log may contain thousands of cases. In this study, an event sequence of a case is formed as an ordered list of task sets and time stamps. A sequence is represented as $((a_1, t_1), (a_2, t_2), (a_3, t_3), \ldots, (a_n, t_n))$, where a_j is a task that occurs with a time stamp which is denoted by t_j, and $1 \leqq j \leqq n$ and $t_{j-1} \leqq t_j$ for $2 \leqq j \leqq n$. To obtain the time-intervals, we proceed to calculate the time Δt between tasks where $\Delta t = t_{i-1} - t_i$ and t_i is a time stamp occurred in the event log. Each time-interval event sequence can be represented as $(a_1, (t_2 - t_1), a_2, (t_3 - t_2), a_3, \ldots, a_{n-1}, (t_n - t_{n-1}))$, where a_i is a task occurred in time stamp t_i in this event sequence, and $1 \leqq i \leqq n$.

After deriving time-intervals between tasks, the max time-interval (*max_time*) and min time-interval (*min_time*) can be found. In addition, a time-interval range R is computed by (*man_ time – min_ time*). We subsequently divide the range R into k time-intervals in which k is set by the user. Therefore, each time-interval has the same range $r = \frac{R}{k}$. Additionally, let a set of discretized time-intervals be represented as $TI = \{ I_0, I_1, I_2, \ldots, I_k, I_\infty \}$, where I_0 denotes the time-interval Δt satisfying $0 \leqq \Delta t \leqq$ *mix_time*, I_1 denotes the time-interval Δt satisfying $min\_time < \Delta t \leqq min\_time + r$, I_j denotes the time-interval Δt satisfying $min\_time + j \times r < \Delta t \leqq min\_time + (j+1) \times r$ for $1 < j < k-1$, I_k denotes the time-interval Δt satisfying $max\_time - r < \Delta t \leqq max\_time$, I_∞ denotes the time-interval Δt satisfying $max\_time < \Delta t \leqq \infty$.

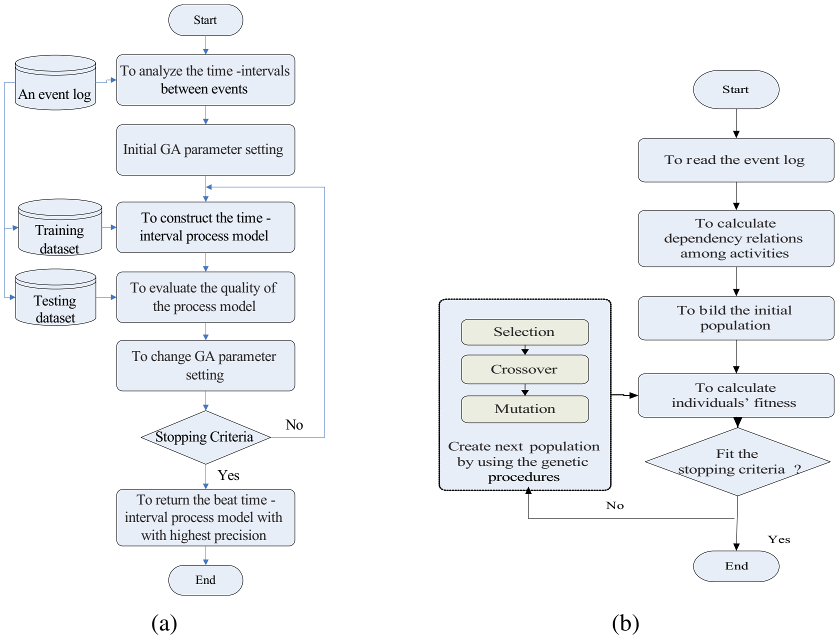

Start
An event log
To analyze the time -intervals between events
Initial GA parameter setting
Training dataset
To construct the time - interval process model
Testing dataset
To evaluate the quality of the process model
To change GA parameter setting
Stopping Criteria
No
Yes
To return the beat time - interval process model with with highest precision
End
(a)
Start
To read the event log
To calculate dependency relations among activities
To bild the initial population
Selection
Crossover
Mutation
Create next population by using the genetic procedures
To calculate individuals' fitness
Fit the stopping criteria ?
No
Yes
End
(b)

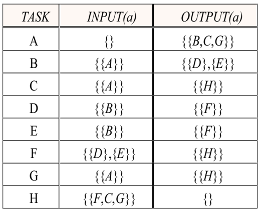
Fig. 1. (a) The framework of the proposed method (b) Genetic process mining procedure

3.2 The Genetic Process Mining Method

To construct process models, this research utilizes a genetic process mining method proposed by [4, 7]. Genetic process mining is the method which uses the genetic algorithm to discover process models from the event logs. Using genetic algorithm to discover process models can overcome the limitation existed in process mining techniques which search answers based on local information such as α-algorithm [1, 7]. The main steps of genetic process mining method are shown in Fig. 1(b) and will be explained as follows.

After deciding population size and number of generation, initial population is composed of random generated individuals. Each individual is encoded using a causal matrix [4]. Fig. 2(a) shows an example causal matrix representing a possible process model in Fig. 2(b). A causal matrix is built with a causality relation *C*. The causality relation represents the relation between two tasks executed from task a_1 to task a_2, where a_1 and a_2 are tasks in the event log. In the causal matrix of an individual, the *INPUT* (*a*) set indicates what tasks are directly preceded by task *a* and the *OUTPUT* (*a*) set indicates what tasks are directly followed by task *a*. Tasks in the *INPUT* (*a*) set and *OUTPUT* (*a*) set show Boolean operations as well. As shown in Fig. 2(a), task *A* is the first task so that *INPUT* (*A*) ={ }. In addition, *OUTPUT* (*A*) = {{*B*, *C*, *G*}} indicates that only one of *B*, *C* or *G* will be executed after task *A* finished.

The fitness measure estimates how an individual fits into the event log. In this research, the fitness measure is divided into two methods. One is "completeness" fitness method $PF_{complete}$ and another is "preciseness" fitness $PF_{precise}$ method. The "completeness" fitness method attempts to obtain the fitness value by comparing each individual (possible process model) with event sequences in the log. In addition, the partial accuracy of an individual and the number of problems occurred in the individual are also calculated in the "completeness" fitness method. The problems occurred in an individual mean that an AND-split expression where should be an OR-split, an OR-join expression where should be an AND-join, an OR-split expression where should be an AND-split and an AND-join where should be an OR-join.



TASK	*INPUT(a)*	*OUTPUT(a)*
A	{}	{{*B,C,G*}}
B	{{*A*}}	{{*D*},{*E*}}
C	{{*A*}}	{{*H*}}
D	{{*B*}}	{{*F*}}
E	{{*B*}}	{{*F*}}
F	{{*D*},{*E*}}	{{*H*}}
G	{{*A*}}	{{*H*}}
H	{{*F,C,G*}}	{}

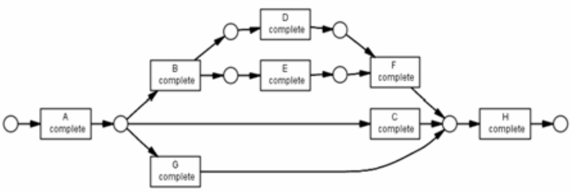


Fig. 2. (a) An example process model (b) The causal matrix of an individual

The stop criteria are used to determine when the algorithm finishes its implementation. In the genetic process mining method, stopping criteria are set as following: (i) an individual with fitness equal to 1 is found; (ii) the maximum generation number is executed; (iii) the best fitness value among half population has not changed. If the stopping criteria are not reached, the genetic operators of the algorithm are used to create a new population.

The evolution theory of Darwin indicates that the best individual should be reserved and should be used to create new offspring. Therefore, the tournament method is used to choose the best individuals for crossover. To generate better individuals, crossover is conducted to combine existing individuals in current population while crossover rate is decided. In crossover procedure, the sets of causality relation in individuals are recombined to generate new individuals. Therefore, the search space contains any combinations of causality relation in all individuals. An individual is allowed to add or lose tasks in the *INPUT* (*a*) and *OUTPUT* (*a*) set, exchange the causality relation with other individuals. First, in crossover procedure, a task *a* is selected randomly to be the *crossover point*. Second, two individuals ($offspring_1$ and $offspring_2$) is copied by parents ($parent_1$ and $parent_2$). Third, a *swap point* is also randomly selected separately from *INPUT* (*a*) set of $offspring_1$ and $offspring_2$, notated sp_1 and sp_2. The *crossover point* (the selected task) will be recombined by exchanging the *remainingSet*(*a*) and *swapSet*(*a*) subsets divided from the *swap point*, where the *remainingSet*(*a*) contains the subsets whose position appear in *INPUT* (*a*) set before the *swap point* and the *swapSet*(*a*)contains the subsets whose position appear in *INPUT* (*a*) from the *swap point* to the end. In a same way, the *swap point*, *remainingSet*(*a*) and *swapSet*(*a*) are executed in the *OUTPUT* (*a*) set of $offspring_1$ and $offspring_2$ also.

Finally, the mutation procedure will insert new characters in parents for creating the variable individuals. Mutation is important to prevent the population from stagnating at any local optimal solutions. In the mutation, the existing causality relations of a population may be changed. Each task of an individual (an offspring) can be mutated by the probability of the mutation rate. The mutation point is a task in an individual.

Therefore, the mutation operations can be applied to *INPUT*(*a*) and *OUTPUT*(*a*) set in the mutation point by randomly selecting one of the following ways to mutation [8]: (i) add a task to the randomly selected subset, (ii) remove a task from the randomly selected subset, (iii) randomly recombine the selected subset.

3.3 The Precision Evaluation for Process Model

It is clear that different genetic parameters such as population size, the number of max generation, crossover rate and mutation rate will generate different process models. Therefore, a method for evaluating the quality of process models constructed with different parameter settings should be defined. For this reason, we define a precision evaluation measure, which is composed of two precision measurements. One measurement is to evaluate the constructed process model using segments in an event sequence, while another is to evaluate the constructed process model using a whole event sequence. The accuracy rate of fitted segments in process model is defined in Equation (1) and the segment precision of a process model is defined in Equation (2).

$$Accuracy = \frac{numParsedSubstring\ (a_1, I_i\ , a_2, PM)}{numSubstring\ (a_1, I_i\ , a_2, PM)} \tag{1}$$

$$PS_{segment} = \frac{SumAccurancyTraces(\ Accuracy\ )}{numTracesLog\ (L_{testing})} \tag{2}$$

Where a_1 and a_2 are the tasks appear in the event sequence, I_i is the discretized time-interval between tasks, *PM* is the process model constructed by the training dataset, $L_{testing}$ is the event sequences in the testing dataset event log, *numParsedSubstring* (a_1, I_i, a_2, *PM*) is the number the substring "a_1 I_i a_2" parsed the constructed process model *PM*, *numSubstring* (a_1, I_i, a_2, *PM*) is the number of the substring "a_1 I_i a_2" in the event sequences of the testing dataset, *Accuracy* is the ratio of the fitted segments in an event sequence, *SumAccuracyTraces*(*Accuracy*) is the summarized accuracy of all event sequences, *numTracesLog* ($L_{testing}$) is the total number of cases in the testing dataset. In addition, the whole sequence precision of a process model is defined in Equation (3).

$$PC_{complete} = \frac{allParsedSequences\ (L_{testing}\ , PM)}{numTracesLog\ (L_{testing})} \tag{3}$$

Where *PM* is the process model constructed by the training dataset, $L_{testing}$ is the event sequences in the testing dataset, *allParesedSequences* ($L_{testing}$,*PM*) is the number of event sequences which are parsed in the constructed process model, *numTracesLog* ($L_{testing}$) is the total number of event sequences in the testing dataset. Finally, the precision evaluation measure is defined in Equation (4).

$$P_{precision} = PS_{segment} \times (1\text{-}\mu) + PC_{complete} \times \mu \tag{4}$$

Where μ is the importance weight for $PC_{complete}$. The range of μ is set at [0, 1]. In addition, the weight of $PS_{segment}$ is suggested to be set as the value higher than the weight of $PC_{complete}$.

4 Implementation and Experimental Result

An amusement park in Taiwan is studied to demonstrate the feasibility of the proposed time-interval process mining method. This theme park includes 4 major areas and 38 amusement facilities. A visitor starts his/her trip from the entrance. The sequence of visiting a set of amusement facilities is regarded as an event sequence. All event sequences and time-interval between events are collected through the questionnaires answered by visitors. For simplification, only the event sequences of Wild West area is appeared in the following discussion. Table 1 shows parts of time-interval event sequences for visitors in the Wild West area.

Table 1. The visitor id and time-interval event sequences of Wild West area

Visitor id	Event sequence
0	(**22**, 20.00, **25**, 25.00,**26**)
1	(**22**, 21.54, **25**, 17.69, **22**, 26.32, **27**)
2	(**25**, 21.11, **22**, 33.57, **27)**
3	(**22**, 16.67, **25**, 20.00, **26**)
4	(**22**, 15.00, **25**, 17.91, **27**)
⋮	⋮
27	(**22**, 14.35, **25**, 16.87, **27**)
28	(**26**, 18.82, **25**, 15.88, **22**, 22.48, **27**)
29	(**22**, 21.11, **25**, 26.67, **26**)

The distribution of time-intervals between amusement facilities can be derived from Table 1 in which the *max_time* is 45.36, *min_time* is 13.33 and range R is 32.03. If we divide the range R into 5 time-intervals (k = 5), therefore, I_0: $0< \Delta t \leqq 13.33$, I_1: $13.33 < \Delta t \leqq 19.736$, I_2: $19.736< \Delta t \leqq 26.142$, I_3: $26.142< \Delta t \leqq 33.548$, I_4: $33.548< \Delta t \leqq 38.954$, I_5: $38.954< \Delta t \leqq 45.36$, and I_6: $45.36< \Delta t \leqq \infty$ where $\Delta t = t_{i+1} - t_i$ and t_i is a timestamp occurred in the sequences. Based on these time-intervals, the event sequences in Table 1 can be translated into the time-interval event sequences. For example, the time-interval event sequence for visitor 1 will be (22, I_2, 25, I_1, 22, I_3, 27).

Subsequently, the k-fold cross-validation method is used to divide the discretized time-interval event sequences into a training dataset and a testing dataset. If k is set as 5, one of these five subsets will be a testing dataset and the remaining four subsets will be the training datasets. Before constructing the process model, the parameters of genetic process mining method should be decided first. In this experiment, the population size is set as 150, the max number generation is set as 800, the crossover rate is set as 0.3, and the mutation rate is set as 0.01. After running the genetic mining algorithm, five process models can be found. Then, the precision evaluation measure of Equation (4) is used to estimate the quality of the five process models. In this experiment, the weight μ

in the precision evaluation measure is set as 0.4 since the $PS_{segment}$ is considered more important than the $PC_{complete}$. Table 2 shows the *Fitness*, $PC_{complete}$, $PS_{segment}$, and $P_{precision}$ of the five process models. It is clear that the model 2 is the best model with highest $P_{precision}$ as shown in Fig. 3.

Table 2. The precisions for the five time-interval process models

model	*Fitness*	$PC_{complete}$	$PS_{segment}$,	$P_{precision}$
1	0.97149	0.83333	0.91667	0.88333
2	0.98310	0.83333	0.94444	0.90000
3	0.96282	0.66667	0.66667	0.66667
4	0.97075	0.66667	0.83333	0.76667
5	0.97384	0.66667	0.86111	0.78333
Average	0.97240	0.73333	0.84444	0.73866

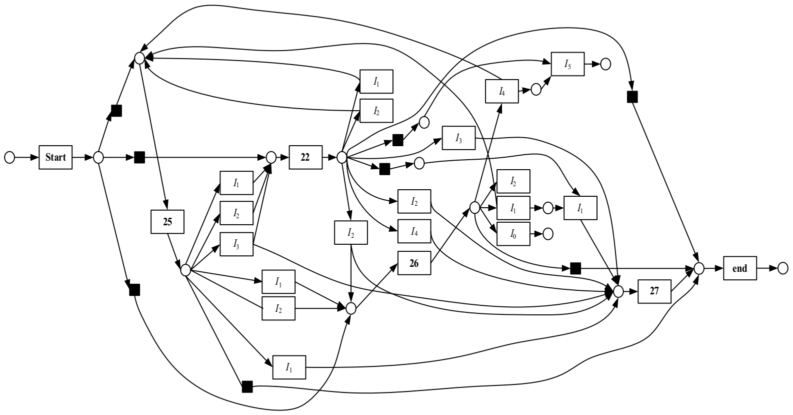


Fig. 3. The best time-interval process model

A process model with time-interval information represents not only event sequences but also the time-intervals between events. In our case, what amusement facilities are visitor's favorites, what the visiting sequence among these amusement facilities, and how long visitors spent between amusement facilities can be clearly observed. For example, as shown in Fig. 3, the time-intervals between amusement facilities from 22 to 27 are I_2 and I_4. Therefore, managers can detect that visitors maybe spend too much time to wait in queue of an amusement facility if most visitors spend time within I_4. If visitors spend long time to wait in queue, managers can arrange up an entertainment program near amusement facility for relieving the boring time when visitors are in queue. Furthermore, the max and min time that visitors spent in the playing sequences can be predicted using the proposed time-interval process model. For example, there are playing sequences such as (Start, 25, $\{I_1, I_2, I_3\}$, 22, $\{I_2, I_4\}$, 27, End), (Start, 22, $\{I_1, I_2\}$, 25, I_2, 27, End) in the model. If a visitor decide to follow the playing sequence from Start, Facility 22, Facility 25, Facility 27 and End, then the min total time for this playing sequence will be 33.066＜total time≦45.878 and the max total time will be

39.482＜ total time ≦52.284. However, if visitors decide to follow the playing sequence from Start, Facility 25, Facility 22, Facility 27 and End, the min total time will be 33.066＜ total time ≦45.878 and the max total time will be 58.69＜ total time ≦ 71.502.

5 Conclusions

Process mining technology plays an important role in constructing a process model of complex process. With process mining technology, the flows of activities (events) can be discovered from the recorded event log. With the constructed process model, managers can understand how activities (tasks) happened in the process. Nowadays, most process mining algorithms consider only the sequences between events, but do not include the time information such as time-stamp and time-interval. This research proposes a time-interval process mining method to overcome the problem. The genetic process mining method is used in our proposed method. Through a proposed precision evaluation measure, a best time-interval process model can be obtained.

The proposed time-interval process model is useful to observe, detect and analyze the behaviors of people or events based on their events sequence and time. In our implementation example, for example, it is clear that this process model help managers easily realize the behaviors of visitors so that managers can provide handy playing sequence recommendation with time information for visitors. The max and min time that visitors spent in the playing sequences can be known beforehand as well. Therefore, visitors can decide their playing sequences if the total visiting time is limited and known in advance. In other aspect such as in marketing, a process model with time-interval can represent the detail buying behaviors of customers. Therefore, managers can provide promotions and customized services for customers at the right time.

References

1. Van der Aalst, W.M.P., van Dongen, B.F.: Discovery workflow performance models from timed logs. In: Han, Y., Tai, S., Wikarski, D. (eds.) EDCIS 2002. LNCS, vol. 2480, pp. 45–63. Springer, Heidelberg (2002)
2. Alves de Medeiros, A.K., van Dongen, B.F., van der Aalst, W.M.P., Weijters, A.J.M.M.: Process Mining: Extending the α-algorithm to Mine Short Loops. BETA Working Paper Series (WP 113), Eindhoven University of Technology, Eindhoven (2004)
3. Ren, C., Wen, L., Dong, J., Ding, H., Wang, W., Qiu, M.: A novel approach for process mining based on event types. In: IEEE International Conference on Services Computing (SCC 2007), vol. 4278740, pp. 721–722 (2007)
4. De Medeiros, A.K.A., Weijters, A.J.M.M., Van Der Aalst, W.M.P.: Genetic process mining: An experimental evaluation. Data Mining and Knowledge Discovery 14(2), 245–304 (2007)
5. Van Der Aalst, W., Weijters, T., Maruster, L.: Workflow mining: Discovering process models from event logs. IEEE Transactions on Knowledge and Data Engineering 16(9), 1128–1142 (2004)
6. De Medeiros, A.K.A., Van Der Aalst, W.M.P., Weijters, A.J.M.M.: Workflow Mining: Current Status and Future Directions. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds.) CoopIS 2003, DOA 2003, and ODBASE 2003. LNCS (LNAI), vol. 2888, pp. 389–406. Springer, Heidelberg (2003)

7. Alves de Medeiros, A.K., Weijters, A.J.M.M., van der Aalst, W.M.P.: Using genetic algorithms to mine process models: Representation, operators and results. BETA Working Paper Series (WP 124), Eindhoven University of Technology, Eindhoven (2004)
8. Cook, J.E., Du, Z., Wolf, A.L.: Discovering Models of Behavior for Concurrent Workflows. Computers in Industry 53(3), 297–319 (2004)
9. Agrawal, R., Gunopulos, D., Leymann, F.: Mining Process Models from Workflow Logs. In: Schek, H.-J., Saltor, F., Ramos, I., Alonso, G. (eds.) EDBT 1998. LNCS, vol. 1377, p. 469. Springer, Heidelberg (1998)
10. Cook, J.E., Wolf, A.L.: Discovering models of software processes from event-based data. ACM Transactions on Software Engineering and Methodology 7(3), 215–249 (1998)
11. Cook, J.E., Wolf, A.L.: Event-based detection of concurrency. In: Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering, pp. 35–45 (1998)
12. Van der Aalst, W.M.P., Weijters, A.J.M.M.: Process mining: a research agenda. Computers in Industry 53, 231–244 (2004)
13. De Medeiros, A.K.A., Van Dongen, B.F., Van Der Aalst, W.M.P., Weijters, A.J.M.M.: Process mining for ubiquitous mobile systems: An overview and a concrete algorithm. In: Baresi, L., Dustdar, S., Gall, H.C., Matera, M. (eds.) UMICS 2004. LNCS (LNAI), vol. 3272, pp. 151–165. Springer, Heidelberg (2004)
14. Wen, L., Van Der Aalst, W.M.P., Wang, J., Sun, J.: Mining process models with non-free-choice constructs. Data Mining and Knowledge Discovery 15(2), 145–180 (2007)
15. Van der Aalst, W.M.P., Van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, A.J.M.M.: Workflow mining: A survey of issues and approaches. Data and Knowledge Engineering 47(2), 237–267 (2003)

Robust Singular Spectrum Transform

Yasser Mohammad and Toyoaki Nishida

Nishida-Sumi Laboratory, Department of Intelligence Science and Technology,
Graduate School of Informatics, Kyoto University, Japan
yasser@ii.ist.i.kyoto-u.ac.jp, nishida@i.kyoto-u.ac.jp

Abstract. Change Point Discovery is a basic algorithm needed in many time series mining applications including rule discovery, motif discovery, casual analysis, etc. Several techniques for change point discovery have been suggested including wavelet analysis, cosine transforms, CUMSUM, and Singular Spectrum Transform. Of these methods Singular Spectrum Transform (SST) have received much attention because of its generality and because it does not require ad-hoc adjustment for every time series. In this paper we show that traditional SST suffers from two major problems: the need to specify five parameters and the rapid reduction in the specificity with increased noise levels. In this paper we define the Robust Singular Spectrum Transform (RSST) that alleviates both of these problems and compare it to RSST using different synthetic and real-world data series.

1 Introduction

The research in change point (CP) discovery problem have resulted in many techniques including CUMSUM [1], wavelet analysis [2], inflection point search [3], autoregressive modeling [4], Discrete Cosine Transform, and Singular Spectrum Analysis SST [5]. Most of these methods with the exception of SST either discover a single kind of change (e.g. CUMSUM discovers only mean shifts), require ad-hoc tuning for every time series (e.g. wavelet analysis), or assumes a restricted generation process (e.g. Gaussian mixtures). The main disadvantages of SST though are the sensitivity to noise and the need to specify five different parameters.

The main idea of SST is to use PCA to discover the degree of 'difference' between the past and future signal pattern around every point in the time series and use the difference as the change score for this point.

Many researchers suggested improvements to traditional SST even though most of these suggestions targeted increasing the speed of the algorithm not its accuracy. [6] introduced online SVD and [7] proposed Krylov Subspace Learning. [7] also proposed using the angle between the subspaces associated with the major PCA components of the past and the future to calculate the change score at every point. The main problem of this proposal is the assumption that all eigen vectors are equal in importance which can be an inaccurate assumption if the distribution of the top eigen values is not nearly uniform (a condition that

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 123–132, 2009.

happens most of the time in our experience with real world time series). In this paper we propose a different approach to utilize the information of the eigen values as well as the eigen vectors for finding the change score.

This paper defines the Robust Singular Spectrum Transform (RSST) for discovering change points in time series which reduces the number of parameters required into two parameters rather than five and dramatically increases the specificity of the traditional SST without decreasing its sensitivity. RSST is linear in time and space requirements as SST and adds very small constant increase in the processing time. Moreover speedup techniques like the use of Krylov Subspace Learning suggested in [7] can be directly utilized with RSST. Extensive comparisons between SST and RSST on synthetic data supports the superiority of RSST (see section 4) in both synthetic and real world data.

2 Singular Spectrum Transform

Moskvina-Zhigljavsky [8] used the singular spectrum analysis technique for change detection. The technique is based on the SVD of the Hankel matrix. As SVD can be applied to almost any kind of matrix the algorithm can be applied to various types of time series without any ad-hoc tuning. The essence of the SST transform is to find for every point $x(i)$ the difference between a representation of the dynamics of the few points before it (i.e. $x(i-p) : x(i)$) and the few points after it (i.e. $x(i+g) : x(i+f)$). This difference is normalized to have a value between zero and one and named $x_s(i)$.

The dynamics of the points before and after the current point are represented using the Herkel matrix which is calculated as:

$$H(t) = [seq(t-n), ..., seq(t-1)] \tag{1}$$

where $seq(t) = \{x(t-w+1), ..., x(t)\}^T$

Singular Value Decomposition (SVD) is then used to find the singular values and vectors of the Herkel Matrix by solving:

$$H(t) = U(t)\, S(t)\, V(t)^T \tag{2}$$

where $S(i-1, i-1) \leq S(i,i) \leq (i+1, i+1)$.

Only the first l left singular vectors ($U_l(t)$) are kept to represent the past change pattern as the hyperplane defined by them. [5] showed that this hyperplane encodes the major directions of change in the signal.

A similar procedure is used to find the direction of largest change in the dynamics for the future of the signal by concatenating m overlapping windows of size w starting g points after t according to:

$$r(t+g) = \{x(t+g), ..., x(t+g+w-1)\}^T \tag{3}$$

$$G(t) = [r(t+g), ..., r(t+g+m-1)] \tag{4}$$

The eigen vector $\beta(t)$ corresponding to the direction of maximum change in the future of the signal is found by solving:

$$G(t)\,G(t)^{T}\,u^{g} = \mu u^{g} \tag{5}$$

$$\beta(t) = u^{g}_{m} \tag{6}$$

where $m = \arg\min_{i}(\mu_i)$

If there is no change in the dynamics of the signal, it is expected that $\beta(t)$ will lie in or very near to the hyperplane represented by U_l (e.g. the directions of maximum change in the past). To quantify the discrepancy between $\beta(t)$ and U_l, we find the projection of $\beta(t)$ onto U_l $(\alpha(t))$.

$$\alpha(t) = \frac{U_l^{T}\beta(t)}{\left\|U_l^{T}\beta(t)\right\|} \tag{7}$$

The change score is then calculated as the cosine of the angle between $\alpha(t)$ and $\beta(t)$:

$$x_s(t) = 1 - \alpha(t)^{T}\,\beta(t) \tag{8}$$

The first problem of the SST algorithm is the need to specify five different parameters. [5] has shown that SST is usually robust to wide variations in w. Domain knowledge or visualization can help in finding an appropriate value for n and w. Choosing the rest of the SST parameters (i.e. g, m and specially l) are more problematic as domain knowledge is not very useful in choosing them. One of the contributions of the proposed RSST transform is to automatically determine a sensible value for these three parameters (section 3).

The second problem of SST is the fast degradation of its Specificity when the input signal is noisy specially with constant or zero background signal as will be shown in section 4. The main contribution of the proposed RSST transform is alleviating this limitation which allows accurate change point discovery under very noisy conditions (section 4).

3 Robust Singular Spectrum Transform

The Robust Singular Spectrum Transform is proposed in this paper as a solution to the two main problems of SST detailed in the previous section.

In SST, the parameter g encodes the delay after which we look for the change in the signal. The change of g affects the results only when the g becomes very near to w. Once g becomes near to w the correlation between the SST transform and the ground truth change points degrades sharply in the synthetic data we used. This suggests that any value of $g \ll w$ is enough. For this reason RSST fixes g at zero. The parameter m serves a similar rule as the parameter n which is deciding how deep we look into the future (the past) for changes in dynamics. As RSST symmetrically processes past and future sequences m is set to n. The parameter l is more difficult to choose. SST fixes this parameter to a value specified by the user. In RSST the value of $l\left(t\right)$ is allowed to change from point to point in the time series depending on the complexity of the signal before it. To calculate a sensible value for l we first sort the singular values of $H\left(t\right)$ and find the corner of the accumulated sum of them $\left(l_{inf}\left(t\right)\right)$ [the point at which the tangent to the curve has an angle of $\pi/4$]. The singular vectors with singular values higher than this value are assumed to be caused by the genuine dynamics of the signal while the other directions encode the effect of noise. This dynamic setting of l reduces the effect of noise on the final results as will be shown in the following section.

After choosing the parameters g, m and $l\left(t\right)$, RSST works in the same way as SST for finding past and future patters and calculates $H\left(t\right)$, $G\left(t\right)$, and $U_l\left(t\right)$ similarly.

To find a first guess of the change score around every point, RSST tries to utilize more information from the future Henkel Matrix $\left(G\left(t\right)\right)$ than SST by using the $l_f\left(t\right)$ eigen vectors of $G\left(t\right)G\left(t\right)^T$ with highest corresponding eigen values $\left(\lambda_{1:l_f}\right)$ rather than only the first eigen vector used in SST. The value of $l_f\left(t\right)$ is selected using the same algorithm for selecting $l\left(t\right)$.

$$G\left(t\right)G\left(t\right)^T u^g = \mu u^g \tag{9}$$

$$\beta_i\left(t\right) = u_i^g,\, i \leq l_f \; and \lambda_{j-1} \leq \lambda_j \leq \lambda_{j+1} \; for \, 1 \leq j \leq w \tag{10}$$

Each one of these l_f directions are then projected onto the hyperplane defined by $U_l\left(t\right)$

The projection of $\beta_i\left(t\right)$s and the hyperplane defined by $U_l\left(t\right)$ is then found using:

$$\alpha_i\left(t\right) = \frac{U_l^T\beta_i\left(t\right)}{\left\|U_l^T\beta_i\left(t\right)\right\|}, i \leq l_f \tag{11}$$

The change scores defined by $\beta_i\left(t\right)$s and $alpha_i\left(t\right)$s are then calculated as:

$$cs_i\left(t\right) = 1 - \alpha_i\left(t\right)^T \beta_i\left(t\right) \tag{12}$$

The first guess of the change score at the point t is then calculated as the weighted sum of these change point scores where the eigen values of the matrix $G\left(t\right)$ are used as weights.

$$\hat{x}\left(t\right) = \frac{\sum\limits_{i=1}^{l_f} \lambda_i \times cs_i}{\sum\limits_{i=1}^{l_f} \lambda_i} \tag{13}$$

After applying the aforementioned steps we get a first estimate $\hat{x}\left(t\right)$ of the change score at every point t of the time series. RSST then applies a filtering step to attenuate the effect of noise on the final scores. The main insight of this filter is that the reduction of SST specificity in noisy signals happens in the sections in which noise takes over the original signal in the time series. The response of SST at these sections can be modeled by a random around a high average for uncorrelated white noise. The filter used by RSST discovers these sections in which the average and the variance of $\hat{x}(t)$ remains nearly constant and attenuates them. The guess of the change score at every point is then updated by:

$$\tilde{x}(t) = \hat{x}(t) \times |\mu_a(t) - \mu_b(t)| \times \left|\sqrt{\sigma_a(t)} - \sqrt{\sigma_b(t)}\right| \quad (14)$$

where μ_a and σ_a are the mean and variance of $\hat{x}(t)$ in a subsequence of length w before the point t while μ_b and σ_b are the mean and variance of $\hat{x}(t)$ in a subsequence of length w after the point t.

RSST then keeps only the local maxima of $\tilde{x}(t)$ and normalizes the resulting time series by dividing with its maximum. This normalized signal $x(t)$ represents the final change score of RSST.

4 Comparison between SST and RSST

Fig. 1 shows visually the effect of adding white noise to the signal on the performance of SST and the proposed RSST. For every condition the original signal, response of the SST transform and response of the RSST transform are shown. Fig. 1.a and Fig. 1.b show the response of both SST and RSST to two changes in dynamics of a data series with strong background signal and with no background signals. Both algorithms perform well in this condition. Fig. 1.d and Fig. 1.e show the effect of adding uniform random noise of range -0.5%:0.5% of the peak to peak (P-P) value of the original signal. Even with this very low noise level, the SST transform gives high response in all the locations in which the noise level is higher than the original signal value. The performance of SST transform is specially unacceptable when there is no background signal. The RSST transform on the other hand performs much better under this very low noise level. Fig. 1.d and Fig. 1.e show the effect of adding uniform random noise of range -50%:50% of the peak to peak (P-P) value of the original signal. Again the performance of RSST is visually much better than the performance of SST under this very noisy condition. This difference in performance will be quantified by extensive tests using synthetic data with adjustable noise levels and background signal strength in this section.

To compare the performance of SST and RSST both in speed and accuracy, 5760 different synthetic time series with controlled embedded changes were produced by changing various features of the time series. Every time series was composed of a recurring pattern called the *background signal* with *embedded patterns* embedded at random locations with controllable numbers. Uniform noise was added to the time series before applying the transforms to it. The dependent variables in this experiment were the background signal strength (peak-to-peak value of the *background signal* divided by the peak-to-peak value of the *embedded pattern*), the noise level, and the generating processes. The length of the

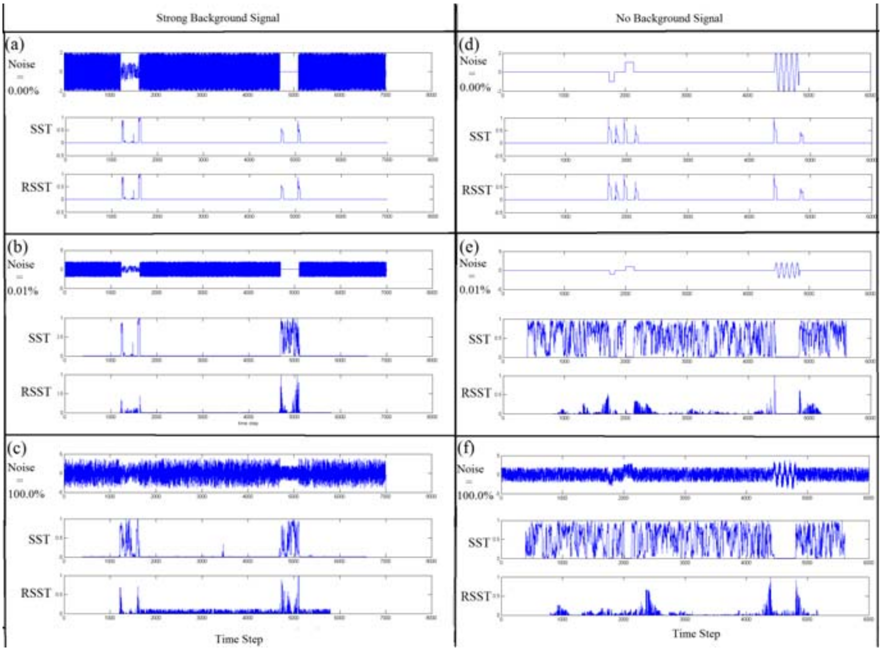


Fig. 1. The Effect of noise Level on the performance of SST and RSST algorithms. For every situation four signals are presented in the following order: Original Time Signal, Ground Truth on the locations of the Change Points, Response of the SST algorithm, Response of the RSST algorithm. Cases (a), (b) and (c) represent increasing noise levels when a strong background signal exists. Cases (d), (e), (f) represent increasing noise levels when no background signal exists.

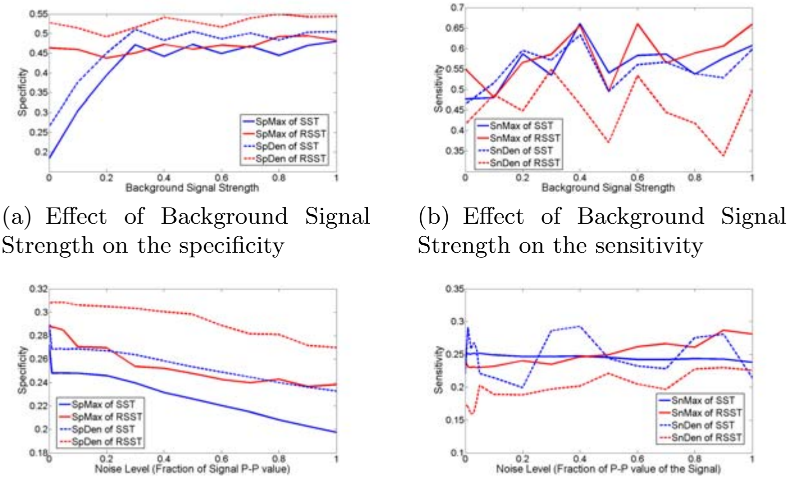
time series was fixed at 8000 points and the length of the embedded pattern was varied from 40 to 400 points. The parameters w, n were fixed at 25 and 10 respectively. Changing w in the range 10 to 60 did not cause any major change in the results. Changing n in the range 5 to 15 also did not cause any major change in the performance. Due to lack of space the results of these tests will not be reported here as [5] have previously achieved similar results for the SST transform. For SST, g and m were selected to equal zero and n respectively (as in the RSST). The value of l was selected to be 3 after experimenting with 300 random samples of the time series and selecting the value that achieved best performance.

For evaluation purposes, we treated the problem as discrete event detection and used specificity and sensitivity to measure the effectiveness of SST and RSST as follows:

1. Sensitivity of Maximum Response *SnMax* defined as the the maximum response of the transform in a window of width $\tau/100$ around the true points of change, averaged over all true points of change. τ is length of the *embedded pattern.*

2. Sensitivity of Response Density *SnDen* defined as the the average response of the transform in a window of width $\tau/10$ around the true points of change, averaged over all true points of change.
3. Specificity of Maximum Response *SpMax* defined as the the maximum response of the transform in all $\tau/10$ windows of the time series excluding the $\tau/10$ windows around the true change points.
4. Specificity of Response Density *SpDen* defined as the the average response of the transform in all the points of the time series excluding all $\tau/10$ windows around the true change points.

Fig. 2-a shows the effect of background signal strength on the specificity of both SST and RSST. As the Figure shows the specificity of RSST is always higher than SST in terms of Specificity of Response Density *SpDen*. This means that RSST is more suitable for applications in which the response around the change points not only the response at the change points is important because RSST will have higher distribution of high responses not only at change points but around them compared with other points in the time series. In terms of Specificity of Maximum Response (SpMax), RSST is also superior for SST both in low and high background signal strengths with a period in which both transforms perform equally between 40% and 70% in our data sets. The important point here is that RSST is not only superior to SST in terms of specificity using both metrics but it is only more stable specially in cases when the background signal strength is low (≤30%). In these cases SST fails to distinguish the changes resulting from the noise and the changes resulting from the original time series because the amplitude of the time series is smaller than the amplitude of the noise. RSST can cope with these situations because of its final filtering step. Fig. 2-b shows



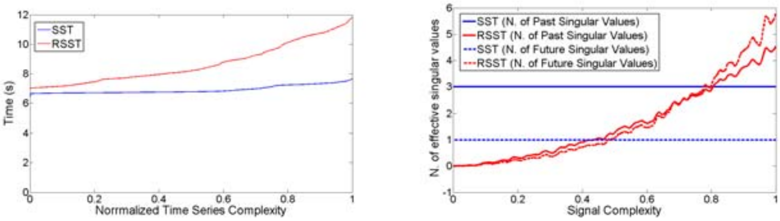
(a) Effect of Background Signal Strength on the specificity

(b) Effect of Background Signal Strength on the sensitivity

(c) Effect of the noise level on the Specificity

(d) Effect of the noise level on the Sensitivity

Fig. 2. Comparison between RSST and SST using synthetic data



(a) Computation time

(b) number of eigen vectors used from the past and future Herkel matrices

Fig. 3. Effect of signal complexity SST and RSST

the effect of background signal strength on the sensitivity of both SST and RSST. As the figure shows there is no clear superiority of one of the transforms over the other in terms of Sensitivity of the Maximum Responses but SST shows slight superiority in the Sensitivity of the Response Density. This can be attributed to the reduced specificity of SST which results in many false high responses, some of them comes naturally very near to the change points increasing the apparent sensitivity of SST. This effect did not happen in the SenMax matrix as the maximum response will not be affected with these noisy false responses.

Fig. 2-c shows the effect of noise level on the specificity of the two transforms. RSST is clearly superior to SST in terms of specificity at all noise levels and the rate degradation of its specificity is less than SST. This superiority is specially clear in the SpDen metric. Fig. 2-d shows the effect of noise level on the sensitivity of the two transforms. Here again SST is slightly superior to RSST due the effect of noisy false responses near the change points explained earlier.

Fig. 3-a shows the effect of signal complexity on the computation times of SST and RSST. Signal Complexity is defined as the multiplication of noise level and number of embedded changes. As the figure shows, the computational complexity of SST and RSST are both linear in signal complexity. RSST has higher slop than SST because as the noise level increases the Specificity of the basic SST transform decreases causing multiple sections of noise-generated high responses that requires more computation to be eliminated (see section 3). Fig. 3-b shows the effect of signal complexity on the number of effective singular values used from the past and future Herkel matrices. SST has fixed values for both of these parameters. RSST on the other hand chooses these parameters automatically and as shown on the figure the number of eigen vectors used are linearly dependent on the signal complexity which is a desirable feature from both computational and accuracy point of view as the size of the matrices used in the projections are kept near the optimal value which attenuates the effect of noise. This may be one of the reasons for the increased specificity and robustness of RSST.

5 Application to Human-Human Interaction Mining

To test the accuracy of RSST on real world data we analyzed the respiration response of 22 participants with ages ranging from 18 to 45 while they were

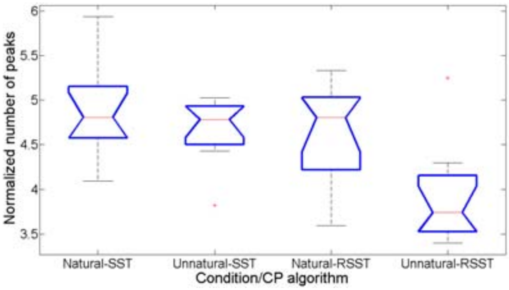


Fig. 4. Number of Peaks in the Change Points detected by SST and RSST

explaining an assembly/disassembly to a listener who either listened carefully (natural session) or did not attend to the explanation (unnatural session). This led to 44 explanation sessions. For more details on this experiment refer to [9]. The respiration signal of the speaker was sampled with rate 100Hz using a Polymate (TEAC Company) device. We then applied RSST and SST to the respiration signal of the instructor. We calculated the number of peaks per minute (change score> 0.6) in the SST and RSST responses in 11 natural and 11 unnatural sessions. Fig. 4 shows a box chart of this count in the two experiment conditions using SST and RSST. Applying t-test no statistical significant difference between the SST response in natural and unnatural conditions was found while the difference in RSST response in these conditions was significant with $p < 0.001$.

6 Conclusions

This paper presented a novel version of Singular Spectrum Transform (SST) called Robust Singular Spectrum Transform (RSST). The main differences between RSST and SST are: 1. fixing the choice of two parameters (g and m). 2. automatically calculating a sensible value for the number of singular vectors representing the past and the future. 3. using multiple future eigen vectors rather than a single one. 4. utilizing a final filtering step to attenuate the effect of noise on the transform. The paper presented extensive evaluation of the RSST algorithm using synthetic data which confirmed its superiority of SST in Specificity and Robustness specially in noisy environments and when the background signal strength of the time series is near zero. The last section of the paper applied the proposed transform to mining physiological data in a controlled human-human interaction experiment and the results show that there is a statistically significant difference in the RSST response to the respiration signal of human subjects when interacting with a listener that behaves naturally and when interacting with a listener that behaves in an un-natural manner. There was no statistically significant difference between these two conditions in the SST response to the same signal.

Directions of future research includes speeding up the calculation of the RSST transform by approximating the SVD calculation and applying it to generate constraints for motif discovery problems that can increase both the speed and accuracy of such algorithms. Extending the algorithm to multidimensional data is another direction of future extensions of this research.

References

1. Basseville, M., Kikiforov, I.: Detection of Abrupt Changes. Printice Hall, Englewood Cliffs (1993)
2. Kadambe, S., Boudreaux-Bartels, G.: Application of the wavelet transform for pitch detection of speech signals. IEEE Transactions on Information Theory 38(2), 917–924 (1992)
3. Hirano, S., Tsumoto, S.: Mining similar temporal patterns in long time-series data and its application to medicine. In: ICDM 2002: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), Washington, DC, USA, p. 219. IEEE Computer Society, Los Alamitos (2002)
4. Gombay, E.: Change detection in autoregressive time series. J. Multivar. Anal. 99(3), 451–464 (2008)
5. Ide, T., Inoue, K.: Knowledge discovery from heterogeneous dynamic systems using change-point correlations. In: Proc. SIAM Intl. Conf. Data Mining (2005)
6. Zha, H., Simon, H.D.: On updating problems in latent semantic indexing. SIAM Journal on Scientific Computing 21(2), 782–791 (1999)
7. Ide, T., Tsuda, K.: Change-point detection using krylov subspace learning. In: Proceedings of the SIAM International Conference on Data Mining (2007)
8. Moskvina, V., Zhigljavsky, A.: An algorithm based on singular spectrum analysis for change-point detection. Communications in Statistics.Simulation and Computation 32(4), 319–352 (2003)
9. Mohammad, Y., Xu, Y., Matsumura, K., Nishida, T.: The h^3r explanation corpus:human-human and base human-robot interaction dataset. In: The fourth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2008) (December 2008)

Hiding Predictive Association Rules on Horizontally Distributed Data

Shyue-Liang Wang[1], Ting-Zheng Lai[3], Tzung-Pei Hong[2], and Yu-Lung Wu[3]

[1] Department of Information Management
[2] Department of Computer Science and Information Engineering
National University of Kaohsiung
Kaohsiung, Taiwan 81148
[3] Institute of Information Management
I-Shou University
Kaohsiung, Taiwan 84001

Abstract. In this work, we propose two approaches of hiding predictive association rules where the data sets are horizontally distributed and owned by collaborative but non-trusting parties. In particular, algorithms to hide the Collaborative Recommendation Association Rules (CRAR) and to merge the (sanitized) data sets are introduced. Performance and various side effects of the proposed approaches are analyzed numerically. Comparisons of non-trusting and trusting third-party approach are reported. Numerical results show that the non-trusting third-party approach has better processing time, with similar side effects to the trusting third-party approach.

Keywords: Privacy preserving, data mining, collaborative recommendation, association rule, horizontally distributed.

1 Introduction

Recent developments in privacy preserving data mining have proposed many efficient and practical techniques for hiding sensitive information that could have been discovered by data mining algorithms. There are four broad areas of research in the field of privacy preserving data mining: privacy preserving data publishing, privacy preserving applications, utility issues, and distributed privacy with adversarial collaboration [1]. In privacy preserving data publishing area, the proposed techniques tend to sanitize the data, by different transformation methods, so that its privacy remains preserved. For example, the k-anonymity methods try to prevent privacy de-identification. In privacy-preserving applications area, it corresponds to designing data management and mining algorithms in such way that the results of association rule or classification rule mining can preserve the privacy of data. A classic example of such technique is association rule hiding, in which some of the association rules are suppressed in order to preserve privacy. The utility-based privacy preservation has two goals: protecting the private information and preserving the data utility as much as possible. In the distributed privacy area, multiple parties may wish to share aggregate private data, without

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 133–141, 2009.

leaking any sensitive information at their end. For example, different companies with sensitive sales data may wish to collaborate among themselves in knowing aggregate trends without leaking the trends of their individual company.

For a single data set, given specific rules or patterns to be hidden, many data altering techniques for hiding association, classification and clustering rules have been proposed. For association rule hiding, three basic approaches have been proposed. The first approach [3,12] hides one rule at a time. The second approach [9,10] deals with groups of restricted patterns or sensitive association rules at a time. The third approach [13,14] deals with hiding certain constrained classes of association rules. Once the proposed hiding items are given, the approach integrates the rule selection process into the hiding process. All these techniques can be considered as *output privacy* hiding on *single data set*. The objective of *output privacy* is that the data set is minimally altered so that the mining result will not disclose certain privacy.

In contrast, the objective of *input privacy* is that the data is manipulated so that the mining result is not affected or minimally affected. For example, the randomization-based techniques [4] and the cryptography-based techniques are some recently proposed techniques for *input privacy*. These techniques can be considered as *input privacy* hiding on *single data set.*

For multiple data sets with adversarial collaboration, the cryptography-based technique such as secure multiparty computation have shown that non-trusting parties can jointly compute functions of their different inputs while ensuring that no party learns anything but the defined output of the function [8,11]. These techniques can be considered as *input privacy* hiding on *distributed data set.*

However, very few works have been proposed on *output privacy* hiding on *distributed data sets.* As demonstrated in the motivating example in this section, non-trusting parties may possess data sets that contain same attributes (horizontally distributed data) and are willing to collaborate and discover patterns from the jointed data set. Nevertheless, they would like to hide certain sensitive association rules before combining the data sets.

In this work, algorithms to hide the collaborative recommendation association rules and to merge the sanitized data sets are introduced. Various characteristics of the proposed algorithms are analyzed. Numerical experiments show that the non-trusting third-party approach performs better than the trusting third-party approach, with similar side effects.

2 Problem Statement

Association rule mining was first introduced in [2]. Let $I = \{ i_1, i_2, \cdots, i_m \}$ be a set of literals, called items. Given a set of transactions *D*, where each transaction T in *D* is a set of items such that $T \subseteq I$, an association rule is an expression $X \Rightarrow Y$ where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \phi$. As an example, for a given database in Table 1, a minimum support of 33% and a minimum confidence of 70%, nine association rules can be found as follows: *B=>A* (66%, 100%), *C=>A* (66%, 100%), *B=>C* (50%, 75%), *C=>B* (50%, 75%), *AB=>C* (50%, 75%), *AC=>B* (50%, 75%), *BC=>A*(50%, 100%), *C=>AB*(50%, 75%), *B=>AC*(50%, 75%), where the percentages inside the parentheses are supports and confidences respectively.

However, mining association rules usually generates a large number of rules, most of which are unnecessary for the purpose of collaborative recommendation. For example, to recommend a target item *{B}* to a customer, the collaborative recommendation association rule set that contains only two rules, *C=>B* (50%, 75%) and *AC=>B*(50%, 75%) will generate the same recommendations as the entire nine association rules found from Table 1. This means that if the new customer has shown interests in purchasing item *{C}* or items *{AC}*, then the collaborative recommender will recommend the new customer to purchase target item *{B}*. Therefore, a collaborative recommendation association rule set can be informally defined as the smallest association rule set that makes the same recommendation as the entire association rule set by confidence priority.

The following is the definition of collaborative recommendation association rule set.

Definition. Let R_A be an association rule set and $R_A^{\,1}$ the set of single-target rules in R_A. A set R_c is a collaborative recommender over R_A if (1) $R_c \subset R_A^{\,1}$, (2) $\forall\ r \in R_c$, there does not exist $r' \in R_c$ such that $r' \subset r$ and $conf(r') > conf(r)$, and (3) $\forall\ r'' \in R_A^{\,1} - R_c$, $\exists\ r \in R_c$ such that $r'' \supset r$ and $conf(r'') \leq conf(r)$.

Several association rule hiding techniques have been proposed in recent years. In this work, we assume that data are horizontally distributed and stored in distributed locations. Given a set of transaction data sets $D_1, D_2, \ldots, D_n$, minimum supports and minimum confidences and sets of recommended items $Y_1, Y_2, \ldots, Y_n$ for each data set, the objective is to minimally modify the data sets $D_1, D_2, \ldots, D_n$, such that no collaborative recommendation association rules containing $Y_1, Y_2, \ldots, Y_n$, on the right hand side of the rule will be discovered from the jointed sanitized data set.

Table 1. Data set D_1

TID	D_1
T_1	ABC
T_2	ABC
T_3	ABC
T_4	AB
T_5	A
T_6	AC

Table 2. Data set D_2

TID	D_2
T_7	BC
T_8	C
T_9	ABC

As an example, for two given horizontally distributed data sets in Table 1 and Table 2, with minimum support of 33%, minimum confidence of 70%, and hidden item $Y = \{C\}$ for both data sets, if transaction T_1 in D_1 is modified from *ABC* to *AB* and transaction T_7 in D_2 is modified from *BC* to *B*, then the following rules that contain item *C* on the right hand side will be hidden: *A=>C* (44%, 57%), *B=>C* (33%, 50%), *AB=>C* (33%, 60%). However, *C=>B* (33%, 60%) will be lost as side effect. The result of the jointed sanitized data set is shown in Table 3.

Table 3. Jointed Data set

TID	D_1+D_2	D_1'+D_2'
T_1	ABC	AB
T_2	ABC	ABC
T_3	ABC	ABC
T_4	AB	AB
T_5	A	A
T_6	AC	AC
T_7	BC	B
T_8	C	C
T_9	ABC	ABC

3 Proposed Approaches and Algorithms

Given a set of horizontally partition data sets, D_1, D_2, ..., D_n, that are owned by non-trusting collaborative parties, if a trusting third party existed, a simple solution to publish jointed but sanitized data set is to submit all data sets to this trusted third party. The third party hides the specified association rules then publishes the sanitized data set, $(D_1 + ... + D_n)'$, this is referred to as *Merge-Then-Hide* (*MTH*) approach or trusting-third-party approach. However, if a trusting third party did not exist, an alternative is to hide designated rules in each data set independently, (D_1', ... , D_n'). The sanitized data sets are then submitted to the third party. The third party then merges the individually sanitized data sets and publishes the results, (D_1' + ... + D_n'), this is referred to as *Hide-Then-Merge* (*HTM*) approach or non-trusting-third-party approach.

To hide an association rule, *X=>Y*, the strategy in the support-based and confidence-based distortion schemes is to either decrease its supports to be smaller than pre-specified minimum support or decrease its confidence to be smaller than pre-specified minimum confidence. In this work, we adopt a similar strategy.

To reduce the number of database scanning, many techniques have been proposed. Based on FP-tree [5] and P-tree [6], we have proposed a similar structure called PI-tree [14]. A PI-tree is an extended prefix-tree structure for storing compressed, crucial information about frequent patterns in less space. A PI-tree is similar to a P-tree except the following. Each node in a PI-tree contains three fields: item name (or item number), number of transactions containing the items on the path from the root to current node (frequency), and a list of transaction ID that contains all the items on the path from the root to current node (TID). For example, the PI-tree for the six transactions in Table 1 is shown in Figure 1. The frequency list is *L* = < *(A:6), (B:4), (C:4)*>.

We proposed two algorithms to perform the two major tasks in the process. The *PI-tree Hiding (PIH)* algorithm builds a PI-tree and sanitizes the transactions to hide collaborative recommendation association rules. The *PI-tree Merging (PIM)* algorithm merges all PI-trees into one PI-tree. The pseudo codes of the proposed algorithms are described as follows.

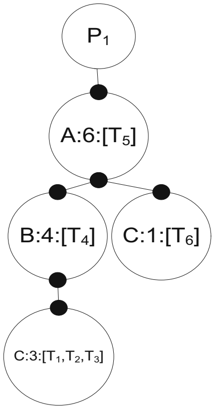


Fig. 1. The PI-tree for data set D_1

Algorithm PI-tree Hiding

Input: (1) a source data set *D*,
(2) minimum support,
(3) minimum confidence,
(4) a set of hidden (recommended) items *Y*,

Output: a sanitized PI-tree *PI'* and frequency list *L'*.

```
1. Build initial PI-tree from D;
  1.1 Loop until no transactions in D;
    1.1.1Read a transaction from D;
    1.1.2Sort according to item name (or item number);
    1.1.3InsertTran(t, PI-Tree);   //insert transaction t to a PI-tree
    1.1.4Update item frequency list L;
  1.2 Sort the updated frequency list L;
2. Restructure the PI-tree
  2.1 Loop until PI-tree is empty;
    2.1.1Extract(path, PI-tree);//extract a path from PI-tree;
    2.1.2Sort(path, L);  //Sort each path according to frequency list L;
    2.1.3InsertPath(path, PI-tree);   //insert a path to a new PI-tree
3. Sanitization //find x -> y and reduce confidence/support
  3.1 Loop until no more hidden item y;
  3.2 For each large 2-itemsets containing y in PI-tree;
    3.2.1Sanitize(y, PI-tree);
4. Output sanitized PI-tree PI' and frequency list L';
```

Algorithm PI-tree Merging

Input: (1) a set of PI-trees,
(2) a set of frequency lists,

Output: a merged PI-tree.

```
1. Combine all frequency lists into L' and sort the list;
2. Sort $PI_i$' trees in descending order according to the highest item frequency in the
      frequency lists of PI trees;
3. Merge all PI-trees
3.1  Loop until each $PI_i$' tree is empty;
3.1.1Extract(path, $PI_i$'), $i >= 2$;     //extract a path
3.1.2InsertPath(path, $PI_1$');        //insert a path to $PI_1$'
4. Restructure the tree according to frequency list L';
4.1  Loop until the $PI_1$' tree is empty
4.1.1Extract(path, $PI_1$')      // extract a path from $PI_1$'
4.1.2Sort(path, L');
4.1.3InsertPath(path, PI');        //insert a path to PI'
5. Return PI';
```

In function *InsertTran([t_1 | T_i], P), a* transaction *t* is sorted into *[t_1 | T_i]* in alphabetical order. Here in each sorted transaction, t_1 is the first item of the transaction and T_i is the set of remaining items in the transaction. The *Extract* function extracts a path from PI-tree. The *Sort* function sorts the nodes in a path according to the frequency list *L*. The *Sanitize* function sanitizes a PI-tree such that rule x -> y is hidden when y is a hidden item, based on the strategy presented in this section.

4 Numerical Experiments

This section reports the numerical performance and characteristics of the proposed *Hide-Then-Merge* and *Merge-Then-Hide* approaches. The artificial data set generated from IBM synthetic data generator [7] is used. The IBM artificial data set has 40,000 transactions, 50 distinct items, with maximum transaction size 11, and average transaction size 5. The minimum supports we tested range from 0.5% to 30%. The minimum confidence range is from 5% to 40%. The total number of association rules ranges from 6 to 404 and the number of hidden rules ranges from 0 to 73. The number of recommended items considered here are one and two items. The experiments are performed on a PC with AMD 1.66 GHz processor and 1 GB RAM running on Windows XP operating system.

To compare the performance and characteristics, we consider the following effects: time effect, data set effect, and side effects. For time effect, we measure the running time required to hide one and two recommending items, i.e., one and two collaborative recommendation association rule sets respectively. For data set effect, we measure the percentage of altered transactions versus all transactions in the original data set (*AT%*). For side effects, we measure the percentages of hiding failure, the new rules generated, and the lost rules respectively. The hiding failure side effect measures the percentage of the number of collaborative recommendation association rules that cannot be hidden over the number of rules to be hidden (*HF%*). The new rule side effect measures the percentage of the number of new rules appeared in the sanitized data set, but not in the original data set, over the number of total association rules in the original data set (*NR%*). This measure quantifies the percentage of the discovered rules that are artifacts. The lost rule side effect measures the percentage of the number of non-sensitive rules that are in the original data set but not in the sanitized data set over the number of association rules in the original data set (*LR%*).

Figure 2 shows the processing times required for *MTH* and *HTM* approaches to hide collaborative recommendation association rule sets for one and two recommended items on IBM artificial data sets. Figure 3 shows the percentages of transactions altered (*AT%*) for *MTH* and *HTM* approaches on IBM artificial data set. Figures 4 and 5 show the hiding failure (HF%), new rule (NR%), and lost rule (LR%) side effects of *MTH* and *HTM* approaches for hiding one recommended item on IBM artificial data set respectively. For *MTH* approach, on the average, there is 2.88% of hiding failures, 1.19% of new rules generated but no lost rules. For *HTM* approach, on the average, there is no hiding failure, 1.05% of new rules generated, and 0.98% of lost rules. Notice that the percentages of hiding failure look relatively high for some data sizes. This is due to the number of rules to be hidden is relatively small. For example, for 5k IBM artificial data set, there are 13 rules to be hidden. But with only one hiding failure, the HF% = 1/13=7.69% looks relatively high comparing to NR% and LR%.

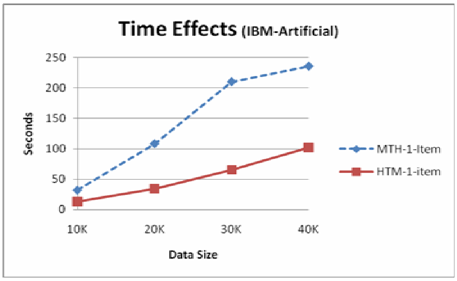


Fig. 2. Time Effects

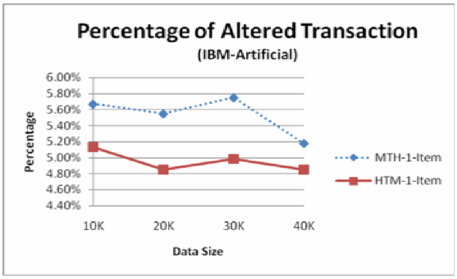


Fig. 3. Data Set Effects

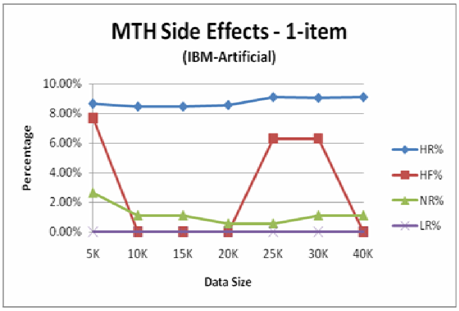


Fig. 4. One-item Side Effects of MTH

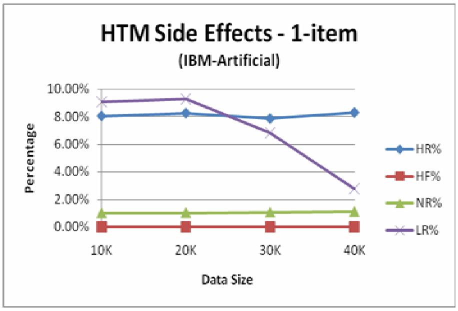


Fig. 5. One-item Side Effects of HTM

In summary, to compare the average performance and effects of the two strategies, we calculate the average processing times, average altered transactions, average hiding failure, average new rules, and average lost rules, of 10k, 20k, 30k, and 40k data sizes. The following observations can be made.

- Processing time: On both data sets, the *HTM* approach requires less processing time to hide collaborate recommendation association rules. This is mainly due to sanitization on two smaller PI-trees requires much less computation time than on larger PI-tree, even with the added cost of merging.

- Data set effect: The difference between average altered transaction percentages *AT%* on IBM data set is 0.09%. To hide two recommended items, the differences are 1.16% and 1.51% respectively. This indicates that number of transactions altered for both approaches are about the same.
- Side effects: Three main effects are measured and each approach shows different strength and weakness. For hiding failure, the proposed *HTM* approach can hide more targeted collaborative recommendation association rules. The effect of creating new non-sensitive rules is about the same for both approaches. However, the proposed *HTM* approach has a drawback of losing more non-sensitive rules than *MTH* approach.

5 Conclusions

Recent development in privacy preserving data mining has proposed many sophisticated data mining techniques that can preserve data and knowledge privacies. However, the problem of preserving output privacy on multiple/distributed data set is not well studied. In this work, we study such a problem and propose *Hide-Then-Merge* and *Merge-Then-Hide* approaches to hide collaborative recommendation association rules from data sets that are horizontally distributed and owned by collaborative but non-trusting parties. This approach is based on the assumption that merging is performed by a non-trusting third party. The results show that the *Hide-Then-Merge* approach requires less processing time but has similar side effects as *Merge-Then-Hide* approach. In the future, we will continue to investigate on hiding other types of rules and patterns and improve data utility of the sanitized data.

Acknowledgments. This work was supported in part by the National Science Council, Taiwan, under grant NSC-97-2221-E-390-021.

References

1. Aggarwal, C., Yu, P.S.: Privacy-Preserving Data Mining: Models and Algorithms. Springer, Heidelberg (2008)
2. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: Proceedings of ACM SIGMOD International Conference on Management of Data, Washington, DC, May 1993, pp. 207–216 (1993)
3. Atallah, A., Bertino, E., Elmagarmid, A., Ibrahim, M., Verykios, V.: Disclosure Limitation of Sensitive Rules. In: Proceedings of IEEE Knowledge and Data Engineering Workshop, Chicago, Illinois, November 1999, pp. 45–52 (1999)
4. Evfimievski, A.: Randomization in Privacy Preserving Data Mining. SIGKDD Explorations 4(2), 43–48 (2002)
5. Han, J., Pei, J., Yin, Y.: Mining Frequent Patterns without Candidate Generation. In: Proceedings of ACM International Conference on Management of Data (SIGMOD), pp. 1–12 (2002)
6. Huang, H., Wu, X., Relue, R.: Association Analysis with One Scan of Databases. In: Proceedings of IEEE International Conference on Data Mining, Maebashi City, Japan, December 2002, pp. 629–632 (2002)

7. IBM. Almaden. Quest Synthetic Data Generation Code, `http://www.almaden.ibm.com/cs/disciplines/iis/`
8. Kantarcioglu, M., Clifton, C.: Privacy-preserving distributed mining of association rules on horizontally partitioned data. In: ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, June 2002, pp. 24–31 (2002)
9. Oliveira, S., Zaiane, O.: An Efficient On-Scan Sanitization for Improving the Balance Between Privacy and Knowledge Discovery. Technical Report TR 03-15, Department of Computing Science, University of Alberta, Canada (June 2003)
10. Oliveira, S., Zaiane, O.: A Unified Framework for Protecting Sensitive Association Rules in Business Collaboration. Int. J. Business Intelligence and Data Mining 1(3), 247–287 (2006)
11. Vaidya, J., Clifton, C.: Privacy Preserving Association Rule Mining in Vertically Partitioned Data. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Canada, July 2002, pp. 639–644 (2002)
12. Verykios, V., Elmagarmid, A., Bertino, E., Saygin, Y., Dasseni, E.: Association Rules Hiding. IEEE Transactions on Knowledge and Data Engineering 16(4), 434–447 (2004)
13. Wang, S.L., Patel, D., Jafari, A., Hong, T.P.: Hiding Collaborative Recommendation Association Rules. Applied Intelligence 26(1), 66–77 (2007)
14. Wang, S.L., Hong, H.P.: One-Scan Sanitization of Collaborative Recommendation Association Rules. In: Proceedings of National Computer Symposium, Taichun, Taiwan, November 2007, pp. 170–176 (2007)

Incremental Mining of Ontological Association Rules in Evolving Environments

Ming-Cheng Tseng[1] and Wen-Yang Lin[2]

[1] Refining Business Division, CPC Corporation, Taiwan
[2] Dept. of Comp. Sci. & Info. Eng., National University of Kaohsiung, Taiwan
clark.tseng@msa.hinet.net, wylin@nuk.edu.tw

Abstract. The process of knowledge discovery from databases is a knowledge intensive, highly user-oriented practice, thus has recently heralded the development of ontology-incorporated data mining techniques. In our previous work, we have considered the problem of mining association rules with ontological information (called ontological association rules) and devised two efficient algorithms, called AROC and AROS, for discovering ontological associations that exploit not only classification but also composition relationship between items. The real world, however, is not static. Data mining practitioners usually are confronted with a dynamic environment. New transactions are continually added into the database over time, and the ontology of items is evolved accordingly. Furthermore, the work of discovering interesting association rules is an iterative process; the analysts need to repeatedly adjust the constraint of minimum support and/or minimum confidence to discover real informative rules. Under these circumstances, how to dynamically discover association rules efficiently is a crucial issue. In this regard, we proposed a unified algorithm, called MIFO, which can handle the maintenance of discovered frequent patterns taking account of all evolving factors: new transactions updating in databases, ontology evolution and minimum support refinement. Empirical evaluation showed that MIFO is significantly faster than running our previous algorithms AROC and AROS from scratch.

Keywords: Association rules, rule maintenance, database update, support constraint refinement, ontology evolution.

1 Introduction

Data mining is to discover useful patterns. One of the most important patterns is to find association rules from a database [1]. An association rule is an expression of the form $X \Rightarrow Y$, where X and Y are sets of items. Such a rule reveals that transactions in the database containing items in X tend to also contain items in Y. And such information is very useful in many aspects of business management, such as store layout planning, target marketing, understanding customer's behavior, etc.

As pointed out in the literature [6], the data mining process is knowledge intensive since it requires domain knowledge to decide what kind of data should be used or which information is the most useful, thus has recently heralded the development of

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 142–151, 2009.

incorporating ontology (domain knowledge) [14] into the process of data mining. In the past few years, there has been researches investigated the problem of mining association rules with classification or composition information [5][7][10][17]. In our recent study, we have considered this problem from a more unified viewpoint. We proposed two effective algorithms, called AROC and AROS [21], for mining association rules that exploit the ontological information not only presenting classification but also composition relationship.

The real world, however, is not static. New transactions are continually added into the database over time, and the ontology of items is evolved accordingly. Furthermore, the analysts would need to repeatedly adjust the constraint of minimum support and/or minimum confidence to discover real informative rules. All of these changes would invalidate previously discovered association rules and/or introduce new ones. Under these circumstances, it is a crucial task to find a method to effectively update the discovered association rules when new transactions are added to the original database along with the ontology evolution and/or minimum support refinement.

In this paper, we consider the problem and propose a unified algorithm, called MIFO, which can handle the maintenance of discovered frequent patterns on taking account of all evolving factors: new transactions updating in databases, ontology evolution, and minimum support refinement. Empirical evaluation showed that MIFO is significantly faster than running our previous algorithms AROC and AROS afresh.

The remainder of this paper is organized as follows. First, related work is presented in Section 2. Section 3 introduces and formalizes the problem for maintenance of association rules with ontological information. In Section 4, we describe the main ideas in the design of the proposed algorithm MFIO. In Section 5, we describe the experiments and discuss the results. Finally, conclusions and future work are given in the last section.

2 Related Work

If concept hierarchy or taxonomy is taken as a kind of ontology, then the research of incorporating ontology into data mining can be traced back to 1995 when Han & Fu [7] and Srikant & Agrawal [17] proposed to combine conceptual hierarchy knowledge to mine so-called *multilevel association rules* and *generalized association rules*, respectively.

In [10], Jea et al. considered the problem of discover multiple-level association rules with composition (*has-a*) hierarchy and proposed a method. Their approach is similar to [17], and was later extended by Chien et al. [5] to fuzzy association rules.

The problem of incremental updating association rules was first addressed by Cheung et al. [2]. By making use of the discovered frequent itemsets, the proposed FUP algorithm can dramatically reduce the efforts for computing the frequent itemsets in the updated database. They further examined the maintenance of multi-level association rules [3], and extended the model to incorporate the situations of deletion and modification [4]. Since then, a number of techniques have been proposed to improve the efficiency of incremental mining algorithm [9][13][15][19]. All of these approaches, however, did not consider the issue of incorporating ontological information. In [20], we have extended the problem of maintaining generalized associations incrementally to that incorporates non-uniform minimum support.

3 Problem Statement

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items, and $DB = \{t_1, t_2, \ldots, t_n\}$ be a set of transactions, where each transaction $t_i = \langle tid, A \rangle$ has a unique identifier *tid* and a set of items A ($A \subseteq I$). To study the mining of association rules with ontological information from *DB*, we assume that the ontology of items, T, is available and is denoted as a graph on $I \cup E$, where $E = \{e_1, e_2, \ldots, e_p\}$ represents the set of extended items derived from I. There are two different types of edges in T, *taxonomic* edge (denoting *is-a* relationship) and *meronymic* edge (denoting *has-a* relationship). We call an item j a *generalization* of item i if there is a path composed of taxonomic edges from i to j, and conversely, we call i a *specialization* of j. On the other hand, we call item k a *component* of item i if there is a path composed of meronymic edges from i to k, and we call i an aggregation of k. For example, in Figure 1(a) I = {HP DeskJet, Epson EPL, Sony VAIO, IBM TP}, and E = {Printer, PC, Ink Cartridge, Photo Conductor, Toner Cartridge, S 60GB, RAM 256MB, IBM 60GB}.

Note that, in Figure 1(a), some item like “Ink Cartridge” could be a component of “HP DeskJet” or be a primitive item purchased directly. To differentiate between these, when an item a serves as a component item, we append an asterisk to the item a (a*), and when a* appears in the association rule, it should be interpreted as “Aggregation (Assembly, Product) of item”. For example, rule IBM TP ⇒ Ink Cartridge* reveals that people who purchase “IBM TP” are likely to purchase the “Product of Ink Cartridge”.

Definition 1. *Given a set of transactions DB and an ontology T, an ontological association rule is an implication of the form,* $A \Rightarrow B$*, where* $A, B \subset I \cup E$, $A \cap B = \varnothing$*, and no item in B is a generalized item or a component of any item in A, and vice versa. The support of this rule,* $sup(A \Rightarrow B)$*, is equal to the support of* $A \cup B$*. The confidence of the rule,* $conf(A \Rightarrow B)$*, is the ratio of* $sup(A \cup B)$ *versus* $sup(A)$*, i.e., the percentage of transactions in DB containing A that also contain B.*

According to Definition 1, an itemset is not only simply composed of primitive purchased items in the ontology but also composed of extended items in higher or lower levels of the ontology. In this regard, we have to take the extended items of items in a transaction into account while determine the support of an itemset. In addition, the condition that no item in A is a generalization or a component item of any item in B is essential; otherwise, a rule of the form, $a \Rightarrow generalization(a)$ or $a \Rightarrow component(a)$, always has 100% confidence and is trivial.

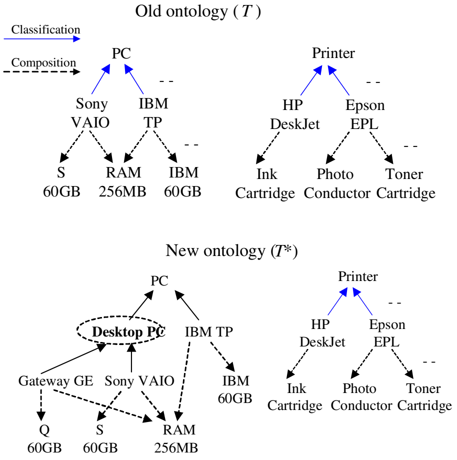
In reality, frequencies of items occur not evenly, that is, some items appear very frequently while others rarely emerge in the transactions. Following the concept in [12], we assume that the user can specify different minimum supports to different items in the ontology, as defined below:

Definition 2. *Let ms(a) denote the minimum support of an item a in $I \cup E$. An itemset $A = \{a_1, a_2, \ldots, a_k\}$, where $a_i \in I \cup E$, is frequent if the support of A is equal to or larger than the lowest value of minimum support of items in A, i.e., $sup(A) \geq ms(A) = \min_{a_i \in A} ms(a_i)$.*

The problem of updating the frequent itemsets with the incremental transactions, the new ontology and the new multiple minimum support setting can be defined as follows.

Definition 3. *Let DB denote the original database, db the incremental database, T the old ontology, T* the new ontology, ms_{old} old multiple minimum support setting, ms_{new} new multiple minimum support setting, A an itemset, and L^{DB} the set of frequent itemsets in DB. The updated extended database UE* is the union of ED* and ed*, i.e., UE* = ED* + ed*, where ED* and ed* are the extension of DB and ed with extended items in T*, respectively. The problem of updating the frequent itemsets with the new ontology T* and the new multiple minimum support setting ms_{new} is, given the knowledge of DB, db, T, T*, L^{DB}, ms_{old} and ms_{new}, to find $L^{UE*} = \{A \mid sup_{UE*}(A) \geq ms_{new}(A)\}$.*

Example 1. Consider Figure 1, which depicts the situation that an original database DB is added with an incremental database db along with the old ontology T replaced by the new ontology T*, where a new primitive item "Gateway GE" along with a new component "Q 60GB" and a new category "Desktop PC" are added, and item "Sony VAIO" is reclassified as "Desktop PC" accordingly. Assume that the analyst has discovered a set of frequent itemsets L^{DB} in DB with respect to the old ontology T and minimum support setting ms_{old}. Then the problem of interest can be regarded as discovering the new set of frequent itemsets L^{UE*} from the updated database UE* with respect to the new minimum support setting.



Original database (*DB*)

TID	Items Purchased
1	IBM TP, Epson EPL, Toner Cartridge
2	Sony VAIO, IBM TP, Epson EPL
3	IBM TP, HP DeskJet, Ink Cartridge
4	HP DeskJet
5	IBM TP, HP DeskJet, Ink Cartridge
6	Sony VAIO, Ink Cartridge

Incremental database (*db*)

TID	Items Purchased
7	Gateway GE, IBM 60GB, Toner Cartridge
8	IBM TP, HP DeskJet, S 60GB, Toner Cartridge
9	Q 60GB, Toner Cartridge
10	IBM 60GB, Toner Cartridge

Fig. 1. An illustration of ontological associations mining in evolving environments

4 Maintenance of Frequent Itemsets with Ontological Information

A straightforward way to find updated ontological frequent itemsets would be to run one of our previous proposed algorithms, AROC or AROS, for finding frequent itemsets in the updated extended database *UE**. This simple way, however, ignores that some of the discovered frequent itemsets are not affected by incremental transactions, minimum support update and/or ontology evolution; that is, these itemsets survive in the ontology evolution and remain frequent in *UE** under the minimum support update. If we can identify the unaffected itemsets, then we can avoid unnecessary computations in counting their supports. In view of this, we adopt an Apriori-like maintenance framework depicted in Figure 2.

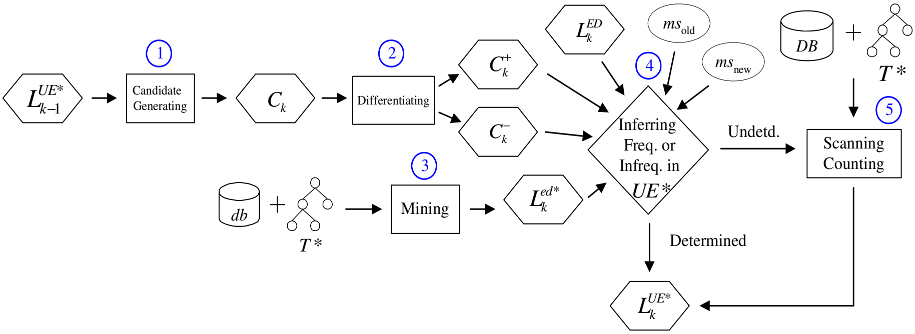


Fig. 2. Proposed framework for updating the frequent ontological *k*-itemsets

Each pass of mining the frequent *k*-itemsets involves the following main steps:

1. Generate candidate *k*-itemsets C_k.
2. Differentiate in C_k the affected itemsets (C_k^+) from the unaffected ones (C_k^-).
3. Scan the incremental database *db* with the new taxonomy T^* , and apply one of our previous algorithms AROC and AROS to find L_k^{ed*} .
4. Incorporate L_k^{ED*} , L_k^{ed*} , C_k^+ , C_k^- , ms_{old} and ms_{new} to determine whether a candidate itemset is frequent or not in the resulting database *UE**.
5. Scan *DB* with T^*, i.e., ED^*, to count the support of itemsets that are undetermined in Step 4.

Below, we elaborate on each of the kernel steps, i.e., Steps 2 and 4.

4.1 Differentiation of Affected and Unaffected Itemsets

Definition 4. *An item (primitive or extended) is called an affected item if its support would be changed with respect to the ontology evolution; otherwise, it is called an unaffected item.*

Consider an item $x \in T \cup T^*$, and the three independent subsets $T - T^*$, $T^* - T$ and $T \cap T^*$. There are three different cases in differentiating whether *x* is an affected item or not. The case that x is a renamed item is neglected for which can be simply regarded as an unaffected item.

1. $x \in T - T^*$. In this case, x is an obsolete item. Then the support of x in the updated database should be counted as 0 and so x is an affected item.
2. $x \in T^* - T$. In this case, x denotes a new item, whose support may change from zero to nonzero. Thus, x should be regarded as an affected item.
3. $x \in T \cap T^*$. This case is more complex, depending on whether x is a primitive, generalized and/or component item, as clarified in the following lemmas.

Lemma 1. *Consider a primitive item $x \in I \cup I^*$. Then $count_{ED^*}(x) = count_{ED}(x)$ if $x \in I \cap I^*$.*

Lemma 2. *Consider an extended item $x \in T \cap T^*$. If $pspc_T(x) \cup pagg_T(x) = pspc_{T^*}(x) \cup pagg_{T^*}(x)$, then $count_{ED^*}(x) = count_{ED}(x)$, where $pspc_T(x)$ and $pspc_{T^*}(x)$ denote the sets of primitive specialized items of x in T and T^*, respectively, and where $pagg_T(x)$ and $pagg_{T^*}(x)$ denote the sets of primitive aggregation items of x in T and T^*, respectively.*

For example, consider the new ontology T^* in Figure 1. The generalized item "PC" is an unaffected item since its primitive specializations, {Gateway GE, Sony VAIO, IBM TP} do not change after the ontology evolution. The component item "RAM 256MB" is also an unaffected item since its primitive aggregations, {Gateway GE, Sony VAIO, IBM TP}, do not change after the ontology evolution.

Definition 5. *For a candidate itemset A, we say A is an affected itemset if it contains at least one affected item.*

Lemma 3. *Consider an itemset A. Then A is an unaffected itemset with $count_{ED^*}(A) = count_{ED}(A)$, if A contains unaffected items only, and $count_{ED^*}(A) = 0$ if A contains at least one obsolete item x, for $x \in T - T^*$, or if A contains at least one new primitive item.*

4.2 Inference of Frequent and Infrequent Itemsets

Now that we have clarified how to differentiate the unaffected and affected itemsets and how to deal with the effect of minimum support refinement, we will further show how to utilize this information to determine in advance whether or not an itemset is frequent before scanning the extended database *ED**, and show the corresponding actions in counting the support of itemsets. Taking account of all three evolving factors, i.e., incremental database update, ontology evolution and/or minimum support refinement, we devise that there are seven situations of itemset updates, as summarized in Table 1.

Note that only Cases 5, 6 and 7 require an additional scan of *ED** to determine the support counts of *A* in *UE**. In Case 3, after scanning *ed** and comparing the support of *A* with ms_{new}, even if *A* is frequent in *ed**, *A* may become infrequent in *UE** under $ms_{new} > ms_{old}$. In Case 4, after scanning *ed** and comparing the support of *A* with ms_{new}, even if *A* is infrequent in *ed**, no matter what ms_{new} is, *A* may become frequent in *UE**. Therefore, only a simple comparison with ms_{new} is required for the support of

A in Cases 3 and 4. In Case 5, *A* is infrequent in *ED*, but frequent in *ed**; therefore, no matter what ms_{new} is, *A* may become frequent in *UE**. After scanning *ed** and comparing the support of *A* with ms_{new}, if *A* is frequent in *ed**, we have to rescan *ED** to decide whether *A* is frequent or not in *UE**. In Case 6, *A* is infrequent in *ED* and *ed**; however, $ms_{new} < ms_{old}$, *A* may become frequent in *UE**. After scanning *ed** and comparing the support of *A* with ms_{new}, even if *A* is infrequent in *ed**, since $ms_{new} < ms_{old}$, we have to rescan *ED** to decide whether *A* is frequent or not in *UE**. For Case 7, since *A* is an affected itemset, its support counts could be changed in *ED**. No matter what ms_{new} is, we need to further scan *ED** to decide whether it is frequent or not.

Table 1. Seven cases for frequent itemsets inference

Conditions				Results		
$T \rightarrow T^*$	L^{ED} (ms_{old})	L^{ed^*} (ms_{new})	ms_{new} to ms_{old}	UE^* (ms_{new})	Action	Case
unaffected	$\in$	$\in$	$\leq$	freq.	no	1
			$>$	undetd.	compare $sup_{UE^*}(A)$ with ms_{new}	3
		$\notin$	$\leq, >$	undetd.	compare $sup_{UE^*}(A)$ with ms_{new}	4
	$\notin$	$\in$	$\leq, >$	undetd.	scan *ED**	5
		$\notin$	$<$	undetd.	scan *ED**	6
			$\geq$	infreq.	no	2
affected	$\in, \notin$	$\in, \notin$	$\leq, >$	undetd.	scan *ED**	7

*Note: Abbreviation “undetd.” represents “undetermined”.

5 Experimental Results

In order to examine the performance of MFIO, we conducted experiments to compare it with that of applying AROC and AROS afresh respect to the new environment. A synthetic dataset of 200000 transactions, generated by the IBM data generator [1], with artificially-built ontology composed of 362 items that are divided into 30 groups, each of which consists of four levels with average fanout of 5.

The efficiency of the algorithms is evaluated from three aspects: the effect of varying minimum support refinement, that of varying incremental size, and that of ontology evolution degree. Here, the evolution degree is measured by the fraction of extended items that are affected by the ontology evolution.

All programs were implemented with the Visual C++ programming language and all experiments were performed on an Intel Pentium-IV 2.80GHz with 2GB RAM, running on Windows 2000. In the implementation of each algorithm, we also adopted two different support counting strategies: one with the horizontal counting [1] and the other with the vertical intersection counting [16]. For differentiation, the algorithms with horizontal counting are denoted as AROC(H), AROS(H) and MFIO(H), while the algorithms with vertical intersection counting are denoted as AROC(V), AROS(V) and MFIO(V). In addition, each performance aspect for MFIO(V) and MFIO(H) was examined under two extreme cases of support refinement: (1) all items having less new minimum supports, i.e., $ms_{new} < ms_{old}$, denoted as MFIO1(V) and MFIO1(H); and (2) all items having larger new minimum supports, i.e., $ms_{new} > ms_{old}$, denoted as MFIO2(V) and MFIO2(H), which can be used as an indication of the performance bounds of our MFIO algorithm.

We first compared the three algorithms under varying incremental size at the multiple minimum support setting (CLS), ms_{old} = CLS + 0.05% for MFIO1 and ms_{old} = CLS − 0.05% for MFIO2 with constant evolution degree 1.8%. The other parameters are set to default values. The results depicted in Figure 3 show that all algorithms exhibit linear scalability, and that MFIO1(H) performs slower than AROS(H) for incremental size over 140,000, because undetermined candidates requiring rescanning the original database increases under $ms_{new} < ms_{old}$, and processing time is proportional to the number of transactions.

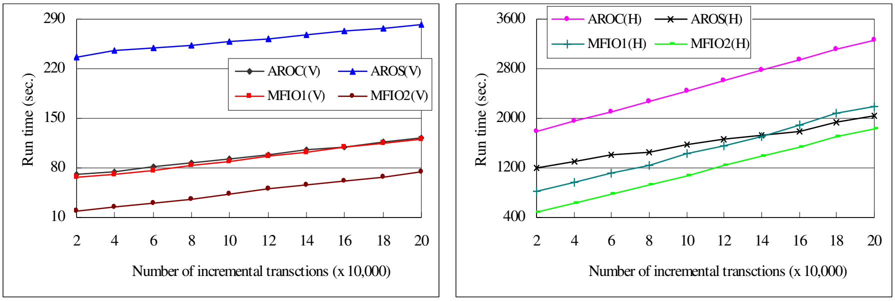


Fig. 3. Performance evaluation for varying incremental transaction size under multiple minimum supports with (a) vertical intersection counting; and (b) horizontal counting

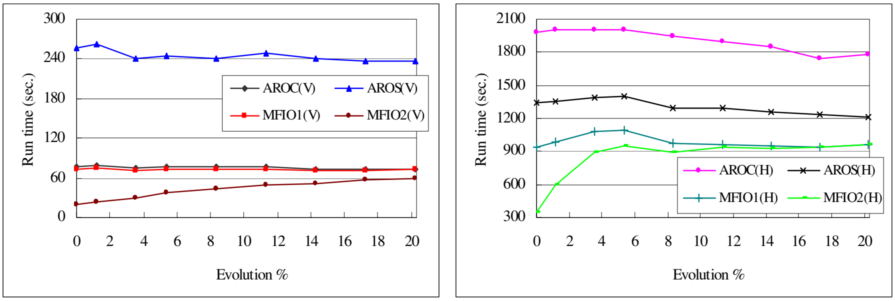


Fig. 4. Performance evaluation for different evolution degrees under multiple minimum supports with (a) vertical intersection counting; and (b) horizontal counting

Next, we examined the effect of varying the degrees of ontology evolution. The other parameters were set as before. As the results in Figure 4 show, algorithm MFIO is heavily affected by the degree of evolution, whereas AROC and AROS exhibit steady performance. When the degree of evolution decreases, the performance gap between MFIO2(V) and AROC(V) and that between MFIO2(H) and AROC(H) increases since the number of affected candidates decreases and so does the possibility of rescanning original database. On the contrary, when the degree of ontology evolution increases, the advantage of reusing original frequent itemsets for MFIO(V) and MFIO(H) disappears gradually. This is because most original frequent itemsets turn out to be affected itemsets, and provoke rescanning the original database.

6 Conclusions and Future Work

In this paper, we have investigated the problem of maintaining discovered association rules with ontological information in an evolving environment. We have considered all possible evolving factors that would affect the feasibility of discovered itemsets, including transaction update, ontology evolution and/or minimum support refinement. To facilitate and accelerate the mining process, we have differentiated affected itemsets from unaffected ones, dealt with minimum support refinement and incremental transactions, and further utilized this information to determine in advance whether or not an itemset is frequent so that this approach can provide another opportunity for candidate pruning and alleviation of database rescan. Finally, we proposed a unified maintaining algorithm, MIFO, for maintaining discovered frequent itemsets. Empirical evaluation showed that the algorithm is very effective and has good linear scale-up characteristic, as compared with applying our previous proposed mining algorithms AROC and AROS from scratch. In the future, we will extend this work to other types of ontology that consists of more complicated semantic relationships [18], and develop other categories of association mining algorithms such as FP-growth [8].

Acknowledgements

This work is partially supported by National Science Council of Taiwan under grant No. NSC95-2221-E-390-024.

References

1. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: Proc. of 20th Int. Conf. on Very Large Data Bases, pp. 487–499 (1994)
2. Cheung, D.W., Han, J., Ng, V.T., Wong, C.Y.: Maintenance of Discovered Association Rules in Large Databases: An Incremental Update Technique. In: Proc. of 12th Int. Conf. on Data Engineering, pp. 106–114 (1996)
3. Cheung, D.W., Ng, V.T., Tam, B.W.: Maintenance of Discovered Knowledge: A Case in Multi-Level Association Rules. In: Proc. of 2nd Int. Conf. on Knowledge Discovery and Data Mining, pp. 307–310 (1996)
4. Cheung, D.W., Lee, S.D., Kao, B.: A General Incremental Technique for Maintaining Discovered Association Rules. In: Proc. of 5th Int. Conf. on Database Systems for Advanced Applications, pp. 185–194 (1997)

5. Chien, B.C., Wang, J.J., Zhong, M.H.: Mining Fuzzy Association Rules on Has-a and Is-a Hierarchical Structures. International Journal of Advanced Computational Intelligence & Intelligent Informatics 11(4), 423–432 (2007)
6. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: The KDD Process for Extracting Useful Knowledge from Volumes of Data. Communications of the ACM 39(11), 27–34 (1996)
7. Han, J., Fu, Y.: Discovery of Multiple-Level Association Rules from Large Databases. In: Proc. of 21st Int. Conf. on Very Large Data Bases, pp. 420–431 (1995)
8. Han, J., Pei, J., Yin, Y.: Mining Frequent Patterns without Candidate Generation. In: Proc. of 2000 ACM SIGMOD Int. Conf. on Management of Data, pp. 1–12 (2000)
9. Hong, T.P., Wang, C.Y., Tao, Y.H.: A New Incremental Data Mining Algorithm Using Pre-Large Itemsets. Intelligent Data Analysis 5(2), 111–129 (2001)
10. Jea, K.F., Chiu, T.P., Chang, M.Y.: Mining Multiple-Level Association Rules in Has-a Hierarchy. In: Proc. of Joint Conf. on AI, Fuzzy System, and Grey System (2003)
11. Lin, W.Y., Tseng, M.C.: Automated Support Specification for Efficient Mining of Interesting Association Rules. Journal of Information Science 32(3), 238–250 (2006)
12. Liu, B., Hsu, W., Ma, Y.: Mining Association Rules with Multiple Minimum Supports. In: Proc. of 1999 Int. Conf. on Knowledge Discovery and Data Mining, pp. 337–341 (1999)
13. Ng, K.K., Lam, W.: Updating of Association Rules Dynamically. In: Proc. of 1999 Int. Symposium on Database Applications in Non-Traditional Environments, pp. 84–91 (2000)
14. OWL Web Ontology Language Use Cases and Requirements (2004), `http://www.w3.org/TR/webont-req/`
15. Sarda, N.L., Srinivas, N.V.: An Adaptive Algorithm for Incremental Mining of Association Rules. In: Proc. of 9th Int. Workshop on Database and Expert Systems Applications, pp. 240–245 (1998)
16. Savasere, A., Omiecinski, E., Navathe, S.: An Efficient Algorithm for Mining Association Rules in Large Databases. In: Proc. of 21st Int. Conf. on Very Large Data Bases, pp. 432–444 (1995)
17. Srikant, R., Agrawal, R.: Mining Generalized Association Rules. In: Proc. of 21st Int. Conf. on Very Large Data Bases, pp. 407–419 (1995)
18. Storey, V.C.: Understanding Semantic Relationships. Very Large Databases Journal 2(4), 455–488 (1993)
19. Thomas, S., Bodagala, S., Alsabti, K., Ranka, S.: An Efficient Algorithm for the Incremental Updation of Association Rules in Large Databases. In: Proc. of 3rd Int. Conf. on Knowledge Discovery and Data Mining, pp. 263–266 (1997)
20. Tseng, M.C., Lin, W.Y.: Maintenance of Generalized Association Rules with Multiple Minimum Supports. Intelligent Data Analysis 8, 417–436 (2004)
21. Tseng, M.C., Lin, W.Y., Jeng, R.: Mining Association Rules with Ontological Information. In: Proc. of 2nd Int. Conf. on Innovative Computing, Information and Control, pp. 300–303 (2007)

Multicriteria Group Decision Support for Information Systems Project Selection

Chung-Hsing Yeh[1], Hepu Deng[2], Santoso Wibowo[2], and Yan Xu[1]

[1] Clayton School of Information Technology, Faculty of Information Technology,
Monash University, Clayton, Victoria 3800, Australia
ChungHsing.Yeh@infotech.monash.edu.au
[2] School of Business Information Technology, RMIT University, GPO Box 2476V,
Melbourne, Victoria 3001, Australia
hepu.deng@rmit.edu.au

Abstract. This paper presents a fuzzy multicriteria group decision making approach for evaluating and selecting information systems projects. The inherent subjectiveness and imprecision of the evaluation process is modeled by using linguistic terms characterized by triangular fuzzy numbers. A new algorithm based on the concept of the degree of dominance is developed to avoid the complex and unreliable process of comparing fuzzy numbers usually required in fuzzy multicriteria decision making. A multicriteria decision support system is proposed to facilitate the evaluation and selection process. An information systems project selection problem is presented to demonstrate the effectiveness of the approach.

Keywords: Information systems project selection, Multicriteria decision making, Fuzzy numbers, Decision support systems.

1 Introduction

The development of information systems (IS) has been actively pursued by organizations in a globalized market to maintain their competitive advantages [1, 2]. The process of evaluating and selecting IS projects for development needs is complex and challenging. It often involves (a) multiple decision makers, (b) multiple selection criteria, and (c) subjective and imprecise assessments [3-5]. To ensure that the best possible IS project is selected with proper justification, it is desirable to use a structured approach capable of comprehensively analyzing the overall performance of available IS projects in a specific decision setting.

As a structured approach, decision support systems (DSS) have been developed to solve various decision problems. For example, Archer and Hasemzadeh [6] develop a DSS for solving the project portfolio selection problem. Bastos et al. [7] apply an intelligent DSS to help solve a resource allocation problem. Ozbayrak and Bell [8] use a rule-based DSS for managing manufacturing parts and tools in a production line. Wen et al. [9] apply an intelligent DSS to analyze decision situations for enterprise acquisition that shows promising results. All of these studies demonstrate that the development of DSS for addressing various decision problems is of great practical benefits. However, in dealing with actual decision settings in real world

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 152–161, 2009.

applications, existing DSSs can be enhanced by (a) addressing the needs of multiple decision makers and multiple criteria, (b) modelling the subjectiveness and imprecision of the human decision making process, and (c) reducing cognitive demand on the decision makers in the process.

To enhance the development of a DSS for assisting decision makers in evaluating and selecting IS projects, in this paper we present an effective fuzzy multicriteria decision making approach. To model the subjectiveness and imprecision of the human decision making process, linguistic terms characterized by triangular fuzzy numbers are used. To avoid the complex and unreliable process of comparing fuzzy numbers usually required in fuzzy multicriteria decision making, a new algorithm is developed by applying the concept of the degree of dominance.

In subsequent sections, we first formulate the IS project selection as a fuzzy multicriteria group decision making problem. We then present a new algorithm for solving the fuzzy multicriteria group decision making problem. A DSS is then proposed and an IS project selection problem is exemplified to demonstrate the applicability of the fuzzy multicriteria decision making approach.

2 Multicriteria Group Decision Making for IS Project Selection

Multicriteria group decision making involves the evaluation of decision alternatives with respect to multiple, often conflicting criteria with the participation of multiple decision makers [3]. Modeled as a multicriteria group decision making problem, the evaluation and selection of IS projects usually involves the following five steps: (a) identifying all the available IS projects as decision alternatives A_i (i = 1, 2, ..., m), (b) determining the evaluation criteria C_j (j = 1, 2, ..., n) and their associated sub-criteria if existent, (c) assessing the performance ratings of the alternatives with respect to each criterion, and the relative importance of the evaluation criteria given by individual decision makers D_k (k = 1, 2, ..., s), (d) aggregating the alternatives' performance ratings and criteria weights for producing an overall preference value for each alternative across all criteria with the inputs from all decision makers, and (e) selecting the most preferred alternative as the most suitable IS project.

To model the uncertainty and subjectiveness present in human decision making, linguistic terms are used to facilitate the subjective assessment to be made by the decision makers. These linguistic terms are represented by triangular fuzzy numbers as their approximate value range between 1 and 9, denoted as (a_1, a_2, a_3), where $1 < a_1 < a_2 < a_3 < 9$. For a linguistic term represented as (a_1, a_2, a_3), a_2 is the most possible value of the term, and a_1 and a_3 are the lower and upper bounds respectively used to reflect the fuzziness of the term. In practical applications, triangular fuzzy numbers are commonly used to characterize linguistic information [10, 11]. The popular use of triangular fuzzy numbers is mainly attributed to their simplicity in both concept and computation. Theoretically, the merits of using triangular fuzzy numbers in fuzzy modeling have been well justified [12]. With the simplest form of the membership function, triangular fuzzy numbers constitute an immediate solution to the optimization problems in fuzzy modeling.

Table 1 shows the linguistic terms (given as in Row 1) and their corresponding triangular fuzzy number for individual decision makers to make qualitative assessments about the performance rating of individual IS project alternatives with

Table 1. Linguistic terms for assessing performance ratings and criteria weights

Linguistic Terms	Very Poor (VP) Very Low (VL)	Poor (P) Low (L)	Fair (F) Medium (M)	Good (G) High (H)	Very Good (VG) Very High (VH)
Membership Function	(1, 1, 3)	(1, 3, 5)	(3, 5, 7)	(5, 7, 9)	(7, 9, 9)

respect to a given criterion. To assess the relative importance of the evaluation criteria, the decision makers can use the linguistic terms given in Row 2 of Table 1, which are characterized by triangular fuzzy numbers as given in Row 3 of Table 1.

3 A Multicriteria Group Decision Making Algorithm

Making multicriteria group decisions is always complex and challenging, due to (a) the uncertainty and imprecision of the human decision making process, (b) the cognitive demand on the decision makers in making subjective assessments, and (c) the comparison of fuzzy numbers which is complex and unreliable [3, 11, 13]. To address these issues, we develop a new algorithm for solving the multicriteria group decision making problem. The proposed algorithm integrates three important concepts in multicriteria decision making research, including (a) the multiattribute value theory (MAVT) [14], (b) the degree of dominance [11], and (c) the ideal solution [10].

With simplicity in both concept and computation, MAVT-based methods are intuitively appealing to the decision makers in practical applications. These methods are particularly suited to decision problems where a cardinal preference or ranking of the decision alternatives is required. In addition, these methods are the most appropriate quantitative tools for group decision support systems [15]. As such, the algorithm developed in this paper can be readily incorporated into a DSS for solving the IS project evaluation and selection problem.

The proposed algorithm starts with assessing the performance rating of each decision alternative A_i (i = 1, 2, ..., m) with respect to each criterion C_j (j = 1, 2, ..., n) by each decision maker D_k (k = 1, 2, ..., s) using the linguistic terms defined as in Table 1. As a result, s decision matrices can be obtained as

$$Y^k = \begin{bmatrix} y_{11}^k & y_{12}^k & \cdots & y_{1n}^k \\ y_{21}^k & y_{22}^k & \cdots & y_{2n}^k \\ \cdots & \cdots & \cdots & \cdots \\ y_{m1}^k & y_{m2}^k & \cdots & y_{mn}^k \end{bmatrix};\ k = 1, 2, \ldots, s \tag{1}$$

where y_{ij}^k is the fuzzy assessment of decision maker D_k about the performance rating of alternative A_i with respect to criterion C_j.

The relative importance of the evaluation criteria C_j can be assessed qualitatively by each decision maker D_k (k = 1, 2, ..., s) using the linguistic terms defined in Table 1. As a result, s fuzzy weight vectors can be obtained as

$$w^k = (w_1^k, w_2^k, \ldots, w_n^k);\ k = 1, 2, \ldots, s \tag{2}$$

By averaging the fuzzy assessments made by individual decision makers as given in (1) and (2), the overall fuzzy decision matrix and the fuzzy weight vector can be obtained as

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \tag{3}$$

$$W = (w_1, w_2, \dots, w_n) \tag{4}$$

where $x_{ij} = \frac{\sum_{k=1}^{s} y_{ij}^{k}}{s}$ and $w_j = \frac{\sum_{k=1}^{s} w_j^{k}}{s}$.

The weighted fuzzy performance matrix that represents the overall performance of each alternative on each criterion can be determined by multiplying the fuzzy criteria weights (w_j) by the alternatives' fuzzy performance ratings (x_{ij}) as

$$Z = \begin{bmatrix} w_1x_{11} & w_2x_{12} & \cdot & w_nx_{1n} \\ w_1x_{21} & w_2x_{22} & \cdot & w_nx_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ w_1x_{m1} & w_1x_{m2} & \cdot & w_nx_{mn} \end{bmatrix} \tag{5}$$

To avoid the unreliable process of comparing fuzzy numbers for determining the overall performance of each alternative across all criteria, the algorithm uses the concept of the degree of dominance between alternatives. The degree of dominance concept is originally used by Yeh and Deng [16] to compare fuzzy numbers *A* and *B* as to how much larger *A* is than *B*. The fuzzy number ranking method based on this concept compares favorably with comparable methods examined. The fuzzy set difference D_{A-B} between *A* and *B* is calculated by fuzzy subtraction as

$$D_{A-B} = A - B = \{(z, \mu_{A-B}(z)), z \in R\} \tag{6}$$

where the membership function of D_{A-B} is defined as

$$\mu_{D_{A-B}}(z) = \sup_{z=x-y} (\min(\mu_A(x), \mu_B(y)), \ x, y \in X). \tag{7}$$

To determine how much larger *A* is than *B*, a defuzzification process is required to extract a single scalar value from D_{A-B}, which can best represent D_{A-B}. Using the mean value of fuzzy numbers method (i.e. the average of value intervals of all α-cuts), the degree of dominance of *A* over *B* is determined by

$$d(A-B) = \int_0^1 D_{A-B}(\alpha)d\alpha \ ; \ D_{A-B}(\alpha) = \begin{cases} (d_{A-B}^{L\alpha} + d_{A-B}^{R\alpha})/2, & 0 \le \alpha \le 1, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

where $d_{A-B}^{L\alpha}$ and $d_{A-B}^{R\alpha}$ are the lower bound and upper bound of the interval $[d_{A-B}^{L\alpha}, d_{A-B}^{R\alpha}]$ respectively, resulting from the α cut on D_{A-B} ($0 \le \alpha \le 1$). *A* dominates *B* if $d(A-B) > 0$, and *A* is dominated by *B* if $d(A-B) < 0$. The larger the value of *d(A-B)*, the higher the degree of dominance of *A* over *B*.

To apply the degree of dominance concept, a common comparison base needs to be established with respect to the weighted fuzzy performance matrix in (5). To achieve this, the concept of the fuzzy maximum and the fuzzy minimum [17] is applied. Given the fuzzy vector (w_jx_{1j}, w_jx_{2j}, …, w_jx_{mj}) of the weighted fuzzy performance matrix for criterion C_j, a fuzzy maximum ($M_{\max}^{j}$) and a fuzzy minimum ($M_{\min}^{j}$) can be determined as in (9) which represent respectively the best and the worst fuzzy performance ratings among all the alternatives with respect to criterion C_j.

$$\mu_{M^j_{\max}}(x) = \begin{cases} \dfrac{x - x^j_{\min}}{x^j_{\max} - x^j_{\min}}, & x^j_{\min} \le x \le x^j_{\max}, \\ 0, & \end{cases} \quad \mu_{M^j_{\min}}(x) = \begin{cases} \dfrac{x^j_{\max} - x}{x^j_{\max} - x^j_{\min}}, & x^j_{\min} \le x \le x^j_{\max}, \\ 0, & \end{cases} \tag{9}$$

where $i = 1, 2,\dots, m$; $j = 1, 2,\dots, n$; $x^j_{\max} = \sup \bigcup_{i=1}^{n} \{x, x \in R$ and $0 < \mu_{w_j x_{ij}}(x) < 1 \}$, and $x^j_{\min} = \inf \bigcup_{i=1}^{n} \{x, x \in R$ and $0 < \mu_{w_j x_{ij}}(x) < 1 \}$.

The degree to which the fuzzy maximum dominates the weighted fuzzy performance ($w_j x_{ij}$) of alternative A_i with respect to criterion C_j can be calculated as

$$d^+_{ij} = d(M^j_{\max} - w_j x_{ij}) = \int D_{(M^j_{\max} - w_j x_{ij})}(\alpha)\, d\alpha \tag{10}$$

$$D_{M^j_{\max} - w_j x_{ij}}(\alpha) = \begin{cases} \dfrac{(d^{L\alpha}_{(M^j_{\max} - w_j x_{ij})} + d^{R\alpha}_{(M^j_{\max} - w_j x_{ij})}}{2}, & 0 \le \alpha \le 1, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Similarly, the degree of dominance of the weighted fuzzy performance ($w_j x_{ij}$) of alternative A_i over the fuzzy minimum with respect to criterion C_j is given as

$$d^-_{ij} = d(w_j x_{ij} - M^j_{\min}) = \int D_{(w_j x_{ij} - M^j_{\min})}(\alpha)\, d\alpha \tag{12}$$

$$D_{(w_j x_{ij} - M^j_{\min})}(\alpha) = \begin{cases} \dfrac{d^{L\alpha}_{(w_j x_{ij} - M^j_{\min})} + d^{R\alpha}_{(w_j x_{ij} - M^j_{\min})}}{2}, & 0 \le \alpha \le 1, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Zeleny [18] first introduces the concept of the ideal solution in multicriteria decision analysis as the best or desired decision outcome for a given decision setting. Hwang and Yoon [13] further extend this concept to include the negative ideal solution in order to avoid the worst decision outcome, known as the technique for order preference by similarity to ideal solution (TOPSIS). This concept has since been widely used in developing multicriteria decision making algorithms for solving practical decision problems, due to (a) its intuitively appealing logic, (b) its simplicity and comprehensibility and (c) its computation efficiency [19-21].

In line with the above concept, the positive fuzzy ideal solution consists of the fuzzy maximum with respect to each criterion across all alternatives, and the negative fuzzy ideal solution consists of the fuzzy minimum in regard to each criterion across all alternatives [16, 22]. The degree of dominance that the positive fuzzy ideal solution is on each alternative A_i and the degree of dominance that each alternative A_i has on the negative fuzzy ideal solution can be calculated respectively as

$$d^+_i = \sum_{j=1}^{n} d^+_{ij} \qquad d^-_i = \sum_{j=1}^{n} d^-_{ij} \tag{14}$$

An alternative is preferred if it is dominated by the positive fuzzy ideal solution by a smaller degree, and at the same time dominates the negative fuzzy ideal solution by a larger degree [16]. Based on this notion, an overall preference value for each alternative A_i across all criteria is calculated as

$$P_i = \frac{(d_i^-)^2}{(d_i^+)^2 + (d_i^-)^2} \tag{15}$$

The larger the preference value P_i, the more preferred the alternative A_i.

4 A Multicriteria Decision Support System

To help the decision makers solve the IS project selection problem in a user-friendly manner, we propose a multicriteria DSS. The DSS will allow the decision makers to input values to express their preferences and assessments and to examine the relationships between the evaluation criteria, and the available alternatives and the selection outcome. Through interaction, the DSS helps the decision makers adopt a problem-oriented approach for solving the IS project selection problem effectively and efficiently.

The proposed DSS is composed of four main components: (a) the data management subsystem, (b) the model base subsystem, (c) the knowledge management subsystem, and (d) the dialogue subsystem. The data management subsystem contains pre-defined connections to internal and external data repositories. This subsystem is responsible for providing data required by other system components. For example, when a decision maker requires specific information about a particular IS project, the data management system will coordinate the acquisition and delivery of the summarized data in the required format. The model base subsystem includes the multicriteria decision making algorithm presented in the previous section. This subsystem may include other analytical tools to analyze and evaluate IS projects. The knowledge management subsystem help the decision makers identify decision alternatives and make assessments. It is inter-connected with the company's knowledge base comprising of IF–THEN rules. The dialogue subsystem provides a user friendly interface for the decision makers to communicate with the DSS.

Using the proposed DSS to select IS projects involves three phases, including (a) pre-evaluation, (b) preference elicitation, and (c) decision analysis and reporting, as shown in Fig. 1. The pre-evaluation phase is used to identify the requirements of the selection problem and to determine the project alternatives. The preference elicitation phase is used to define individual linguistic terms and the corresponding triangular fuzzy numbers, and to determine the criteria weights and performance ratings of project alternatives. In determining the criteria weights, the decision makers can carry out sensitive analysis on weights and examine their effects on the outcome. In practical applications, all the assessments with respect to criteria weights and alternative performance are not always fuzzy. This is because the criteria may include both quantitative and qualitative measures that satisfy the requirements of the selection problem and the judgments of the decision makers [23]. As such, both crisp and fuzzy data are often present simultaneously in a specific multicriteria selection problem [24].

The criteria weight and performance ratings of project alternatives can be assessed by a crisp value or using a linguistic term, depending on the preference or judgment of the decision makers. To maintain the consistence of assessment data in both crisp and fuzzy forms, the decision makers' quantitative assessments are made using a crisp value in the range of 1 to 9. To make qualitative assessments, the decision makers use a set of linguistic terms. The decision makers can use the default settings given in Table 1 or define their own term set from the universe U = {excellent, very high, high to very high, high, fairly high, medium, fairly low, low, low to very low, very low, none}, which is available from the knowledge base of the DSS. The decision makers also have the option of defining the value range or the membership function of triangular fuzzy numbers to be used for representing the linguistic terms in their assessments. The DSS enables the decision makers to make both quantitative and qualitative assessments, because the multicriteria decision making algorithm developed in this paper for solving the IS project selection problem can handle both crisp and fuzzy assessment data.

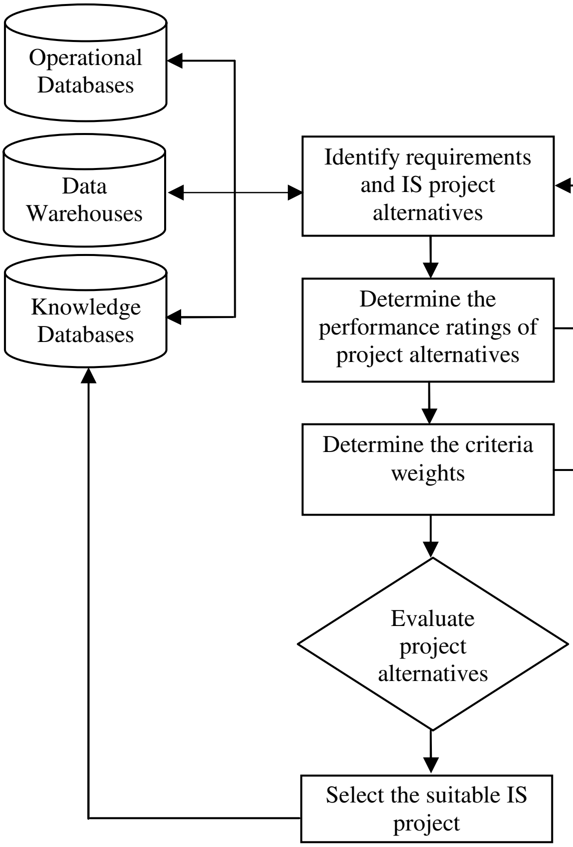

Operational Databases
Data Warehouses
Knowledge Databases
Identify requirements and IS project alternatives
Determine the performance ratings of project alternatives
Determine the criteria weights
Evaluate project alternatives
Select the suitable IS project

Fig. 1. The DSS framework for IS project evaluation and selection

In the decision analysis and reporting phase, the multicriteria decision making algorithm is applied to evaluate and select the most suitable IS project. The overall preference value of each IS project alternative, relative to other project alternatives, is obtained by aggregating the criteria weights and its performance ratings using the algorithm. Based on the overall preference value and ranking of all project alternatives, the most suitable IS project alternative can be recommended in a rational and justifiable manner.

5 An IS Project Selection Problem

To demonstrate the applicability of the multicriteria decision making algorithm, an IS project selection problem at a high-technology manufacturing company is presented. The problem starts with the formation of a committee consisting of three decision makers (D_1, D_2 and D_3) for selecting among three IS project alternatives (A_1, A_2 and A_3). A Delphi process is used to determine a set of selection criteria which meet the requirements of the selection problem. The process helps prioritize the criteria and reaches a consensus about the important criteria for evaluating and selecting the available IS projects. As a result of the process, five evaluation areas in relation to the requirements of the problem are identified, including Organizational Alignment, Potential Risk, Financial Attractiveness, Vendor Characteristics, and System Flexibility.

Organizational Alignment (C_1) reflects the perception of the decision makers on how individual IS project alternatives serve the overall long-term business strategy and organizational objectives in a competitive environment. This is measured by the contribution to the organizational objectives (C_{11}), importance to the organizational profile (C_{12}), relevance to critical success factors (C_{13}), and aid for organizational competitiveness (C_{14}).

Potential Risk (C_2) is related to the potential of failure of the IS project. Issues such as the technical risk (C_{21}), the risk of cost overruns (C_{22}), and the size risk of individual project (C_{23}) are taken into account.

Financial Attractiveness (C_3) is concerned with the economical feasibility of the IS project with respect to the resource limitation of the organization and its business strategy [1]. It is measured by the project cost (C_{31}), the contribution of the project to organizational profitability (C_{32}), and the project's growth rate (C_{33}).

Vendor Characteristics (C_4) is concerned with the decision maker's confidence in vendors' products and services to the organization. Issues such as the vendor's ability (C_{41}), the implementation and maintenance (C_{42}), the consulting service (C_{43}), and the vendor's reputation (C_{44}) are considered.

System Flexibility (C_5) is about the system capabilities in its responsiveness and adaptability to accommodate the organization's requirements and needs [25]. This is measured by the upgrade capability (C_{51}), the ease of integration (C_{52}), and the ease of in-house development (C_{53}).

The 17 measures identified for the five evaluation areas are the evaluation criteria for the selection problem. These evaluation criteria are independent of each other, thus suitable for use in the MAVT-based multicriteria decision making algorithm developed in this paper.

Using the linguistic terms defined in Table 1, the performance of three IS project alternatives (A_1, A_2 and A_3) with respect to the 17 evaluation criteria are assessed by the three decision makers (D_1, D_2 and D_3). Columns 2-10 of Table 2 show the assessment results, which constitute three fuzzy decision matrices as given in (1). Using the linguistics terms defined in Table 1, the importance of the 17 criteria is assessed by the three decision makers. Columns 11-13 of Table 2 show the assessment results, which constitute three fuzzy weight vectors as given in (2).

The overall preference value of each project alternative across all the criteria can be obtained by applying the algorithm given in (3)-(15) to the decision matrices and weight vectors given in Table 2. Table 3 shows the result, which suggest that A_1 is the most suitable project alternative.

Table 2. Performance assessments of alternative IS projects and criteria weights

Criteria	Project A_1			Project A_2			Project A_3			Criteria weight		
	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3	D_1	D_2	D_3
C_{11}	VG	G	VG	VG	VG	VG	G	VG	VG	VH	VH	VH
C_{12}	G	VG	VG	VG	VG	G	G	G	VG	VH	VH	VH
C_{13}	VG	VG	VG	G	VG	VG	G	VG	VG	VH	VH	H
C_{14}	VG	G	G	VG	G	VG	VG	G	G	VH	VH	VH
C_{21}	VG	VG	G	VG	VG	G	G	G	G	VH	H	VH
C_{22}	G	VG	G	VG	VG	G	G	VG	VG	VH	H	VH
C_{23}	G	VG	VG	G	VG	VG	VG	VG	VG	VH	VH	VH
C_{31}	VG	VG	VG	G	VG	VG	G	VG	G	VH	VH	H
C_{32}	G	VG	VG	G	VG	VG	G	VG	G	VH	VH	VH
C_{33}	VG	G	VG	VG	G	VG	G	VG	VG	VH	VH	H
C_{41}	G	VG	G	G	G	G	G	G	G	H	H	VH
C_{42}	VG	VG	G	G	VG	G	G	G	G	VH	VH	VH
C_{43}	G	VG	G	G	VG	VG	VG	G	G	H	VH	VH
C_{44}	VG	G	G	G	G	VG	VG	VG	VG	H	H	VH
C_{51}	VG	G	VG	VG	G	VG	G	VG	VG	VH	H	VH
C_{52}	G	VG	G	VG	G	G	VG	G	G	VH	VH	VH
C_{53}	VG	VG	G	G	VG	G	G	VG	G	VH	VH	H

Table 3. The overall preference value and ranking of IS project alternatives

IS Project Alternative	Preference value	Ranking
A_1	0.89	1
A_2	0.74	2
A_3	0.62	3

6 Conclusion

Evaluating and selecting IS projects is a complex process, as it involves multiple decision makers making subjective and imprecise assessments in relation to multiple decision alternatives and evaluation criteria. To address this complex issue, we have formulated the selection problem as a fuzzy multicriteria group decision making problem and developed an effective algorithm for solving the problem. The merits of the algorithm developed facilitate its incorporation into a group decision support system for solving practical IS project selection problems. With its simplicity in concept and computation, the algorithm is applicable to the general multicriteria evaluation and selection problem involving fuzzy assessments.

References

1. Deng, H., Wibowo, S.: A Fuzzy Multicriteria Analysis Approach Using Pairwise Comparison to Selecting Information Systems Projects under Uncertainty. In: 19th International Conference on Multiple Criteria Decision Making, Auckland, N.Z. (2008)
2. Stewart, R.A.: A Framework for the Life Cycle Management of Information Technology Projects: ProjectIT. Int. J. Proj. Manage. 26, 203–212 (2008)

3. Shih, H.S., Huang, L.C., Shyur, H.J.: Recruitment and Selection Processes through an Effective GDSS. Comput. Maths. Appl. 50, 1543–1558 (2005)
4. Ghasemzadeh, F., Archer, N.P.: Project Portfolio Selection through Decision Support. Decis. Support Syst. 29, 73–88 (2000)
5. Lee, J.W., Kim, S.H.: An Integrated Approach for Interdependent Information System Project Selection. Int. J. Proj. Manage. 19, 111–118 (2001)
6. Archer, N.P., Hasemzadeh, F.: Project Portfolio Selection through Decision Support. Decis. Support Syst. 29, 73–88 (2000)
7. Bastos, R.M., de Oliveira, F.M., de Oliveira, J.P.M.: Autonomic Computing Approach for Resource Allocation. Exp. Syst. Appl. 28, 9–19 (2005)
8. Ozbayrak, M., Bell, R.: A Knowledge-Based Support System for the Measurement of Parts and Tools in FSM. Decis. Support Syst. 35, 487–515 (2003)
9. Wen, W., Wang, W.K., Wang, T.H.: A Hybrid Knowledge-Based Decision Support System for Enterprise Mergers and Acquisitions. Exp. Syst. Appl. 28, 569–582 (2005)
10. Yeh, C.-H., Deng, H., Chang, Y.-H.: Fuzzy Multicriteria Analysis for Performance Evaluation of Bus Companies. Eur. J. Oper. Res. 126, 459–473 (2000)
11. Deng, H.: Multicriteria Analysis with Fuzzy Pairwise Comparison. Int. J. Approx. Reasoning 21, 215–231 (1999)
12. Pedrycz, W.: Why Triangular Membership Functions? Fuzzy Sets Syst. 64, 21–30 (1994)
13. Hwang, C.L., Yoon, K.S.: Multiple Attribute Decision making: Methods and Applications. Springer, Berlin (1981)
14. Keeney, R., Raiffa, H.: Decisions with Multiple Objectives, Preferences and Value Tradeoffs. Cambridge University Press, New York (1993)
15. Matsatsinis, N.F., Samaras, A.P.: MCDA and Preference Disaggregation in Group Decision Support Systems. Eur. J. Oper. Res. 130, 414–429 (2001)
16. Yeh, C.-H., Deng, H.: A Practical Approach to Fuzzy Utilities Comparison in Fuzzy Multicriteria Analysis. Int. J. Approx. Reasoning 35(2), 179–194 (2004)
17. Chen, S.H.: Ranking Fuzzy Numbers with Maximizing Set and Minimizing Set. Fuzzy Sets Syst. 17, 113–129 (1985)
18. Zeleny, M.: Multiple Criteria Decision Making. McGraw-Hill, New York (1982)
19. Deng, H., Yeh, C.-H., Willis, R.J.: Inter-Company Comparison Using Modified TOPSIS with Objective Weights. Comput. Oper. Res. 27, 963–973 (2000)
20. Wang, Y.J., Lee, H.S.: Generalizing TOPSIS for Fuzzy Multiple-Criteria Group Decision-Making. Comput. Maths. Appl. 53, 1762–1772 (2007)
21. Yeh, C.-H., Chang, Y.-H.: Modeling Subjective Evaluation for Fuzzy Group Multicriteria Decision Making. Eur. J. Oper. Res. 194, 464–473 (2009)
22. Chang, Y.-H., Yeh, C.-H.: A New Airline Safety Index. Transport. Res. B-Meth. 38, 369–383 (2004)
23. Deng, H., Wibowo, S.: A Rule-Based Decision Support System for Evaluating and Selecting IS Projects. In: IAENG International Conference on Operations Research (ICOR 2008), Hong Kong (2008)
24. Deng, H.: A Fuzzy Approach to Selecting Information Systems Projects. Int. J. of Comput. Inform. Sci. 6, 13–21 (2005)
25. Lin, C.T., Hsu, P.F., Sheen, G.J.: A Fuzzy-Based Decision-Making Procedure for Data Warehouse System Selection. Exp. Syst. Appl. 32, 939–953 (2007)

Utility-Based Repair of Inconsistent Requirements

Alexander Felfernig[1], Markus Mairitsch[2], Monika Mandl[1],
Monika Schubert[1], and Erich Teppan[2]

[1] Applied Software Engineering, Graz University of Technology,
Inffeldgasse 16b, A-8010 Graz, Austria
{alexander.felfernig,monika.mandl,
monika.schubert}@ist.tugraz.at
[2] Intelligent Systems and Business Informatics, University Klagenfurt,
Universitaetsstrasse 65-67, A-9020 Klagenfurt, Austria
{markus.mairitsch,erich.teppan}@uni-klu.ac.at

Abstract. Knowledge-based recommender applications support the customer-individual identification of products from large and complex assortments. Recommendations are derived from customer requirements by interpreting filter constraints which reduce the set of possible products to those relevant for the customer. If no solution could be found for the requirements, repair actions are proposed which support customers in finding a way out of the "no solution could be found" dilemma. State-of-the-art systems support the identification of repair actions based on minimality assumptions, i.e., repair alternatives with low-cardinality changes are favored compared to alternatives including a higher number of changes. Consequently, repairs are calculated using breadth-first conflict resolution which not necessarily results in the most relevant changes. In this paper we present the concept of utility-based repairs which integrates utility-based recommendation with efficient conflict detection algorithms and the ideas of model-based diagnosis (MBD).

Keywords: Knowledge-based Recommenders, Utility-based Repair, Diagnosis.

1 Introduction

Knowledge-based recommenders support users in the identification of interesting products from large and complex assortments. Compared to the wide-spread and also well-known recommendation approaches of content-based filtering [11] and collaborative filtering [9], those recommenders rely on an explicit representation of the product properties and the corresponding recommendation rules [1][4]. Knowledge-based recommenders are characterized by two knowledge sources: knowledge about the *customer requirements* and knowledge about the *product domain*. Furthermore, *constraints* relate customer requirements to the corresponding product properties. Consequently, in the line of [7] we focus on the application of a specific type of knowledge-based recommendation: constraint-based recommendation. Users interacting with a knowledge-based recommender typically have to answer questions (conversational recommendation) within the scope of a dialog [5]. The dialog can be modeled either explicitly in the form of, for example, a finite state automaton or implicitly by letting users select

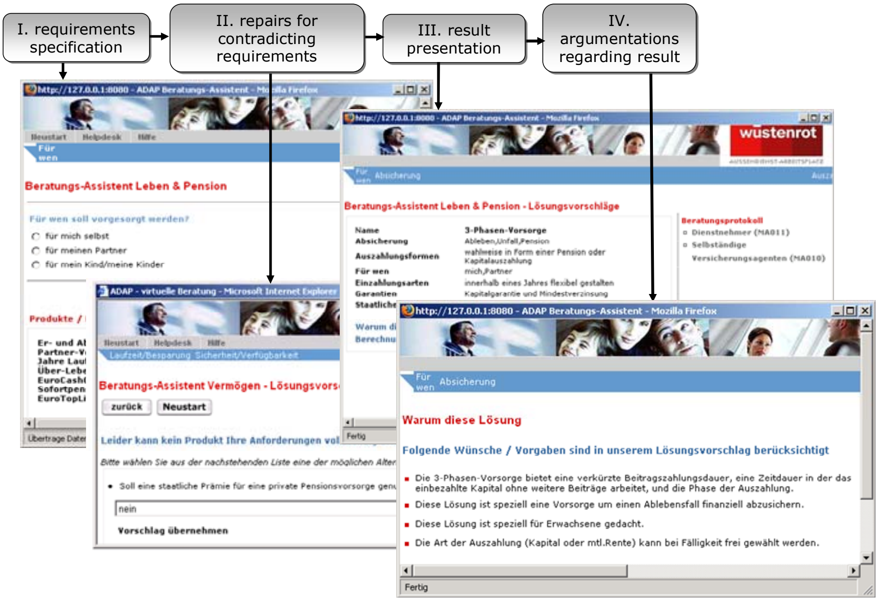
B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 162–171, 2009.

the most interesting attributes on their own [7]. If no product fits the given set of requirements, those recommenders typically support the user in getting out of the so-called “no solution could be found” dilemma.

An example for a commercially available knowledge-based recommender application is depicted in Fig. 1. Our team has deployed this application for the Wüstenrot building and loan association in Austria as a supportive tool for the existing Customer Relationship Management (CRM) environment. A typical recommendation process consists of four steps. First, the user is engaged in a requirements elicitation process where all the relevant requirements are collected in order to be able to calculate a recommendation (Phase I.). If no solution could be found, the recommender proposes a set of repair alternatives which could be selected by the user in order to be able to find a solution (Phase II.). Then, the set of possible solutions is presented (Phase III.). Finally, for each solution, an explanation (set of arguments) is calculated which explain the reason as to why the product has been selected (Phase IV). Such explanations play an important role since they help to increase the quality of recommenders in terms of trust in recommended products and satisfaction with the application.

In this paper we focus on the *calculation of repair actions* (Phase II.) which support the automated and personalized recovery from dead-ends. Knowledge-based recommenders support this phase by proposing minimal cardinality sets of repair actions s.t. a solution could be found. However, such approaches do not guarantee that the repair actions fit to the wishes and needs of the customer.

In this paper we contribute to the advancement of the state-of-the-art in interactive systems by presenting a new approach to the *repair of inconsistent requirements*. With this, we prevent users from spending too much time to find a way out from the


I. requirements specification
II. repairs for contradicting requirements
III. result presentation
IV. argumentations regarding result
http://127.0.0.1:8080 - ADAP Beratungs-Assistent - Mozilla Firefox
Neustart
Helpdesk
Hilfe
Für wen
Übertrage Daten
ADAP - virtuelle Beratung - Microsoft Internet Explorer
Laufzeit/Besparung
Sicherheit/Verfügbarkeit
zurück
Neustart
Vorschlag übernehmen
wüstenrot
Absicherung
Fertig

Beratungs-Assistent Leben & Pension

Für wen soll vorgesorgt werden?

○ für mich selbst
○ für meinen Partner
○ für mein Kind/meine Kinder

Produkte /

Er- und Al
Partner-V
Jahre Lau
Über-Lebe
EuroCash
Sofortpen
EuroTopLi

Beratungs-Assistent Vermögen - Lösungsvors

Leider kann kein Produkt Ihre Anforderungen vol

Bitte wählen Sie aus der nachstehenden Liste eine der möglichen Alter

- Soll eine staatliche Prämie für eine private Pensionsvorsorge genu

nein

Beratungs-Assistent Leben & Pension - Lösungsvorschläge

Name	3-Phasen-Vorsorge
Absicherung	Ableben,Unfall,Pension
Auszahlungsformen	wahlweise in Form einer Pension oder Kapitalauszahlung
Für wen	mich,Partner
Einzahlungsarten	innerhalb eines Jahres flexibel gestalten
Garantien	Kapitalgarantie und Mindestverzinsung
Staatliche	

Warum di
Berechnu

Beratungsprotokoll
- Dienstnehmer (MA011)
- Selbständige Versicherungsagenten (MA010)

Warum diese Lösung

Folgende Wünsche / Vorgaben sind in unserem Lösungsvorschlag berücksichtigt

- Die 3-Phasen-Vorsorge bietet eine verkürzte Beitragszahlungsdauer, eine Zeitdauer in der das einbezahlte Kapital ohne weitere Beiträge arbeitet, und die Phase der Auszahlung.
- Diese Lösung ist speziell eine Vorsorge um einen Ablebensfall finanziell abzusichern.
- Diese Lösung ist speziell für Erwachsene gedacht.
- Die Art der Auszahlung (Kapital oder mtl.Rente) kann bei Fälligkeit frei gewählt werden.

Fig. 1. Example Recommender Application

“no solution could be found” dilemma. In Section 2 we introduce a working example from the domain of digital cameras. In Section 3 we introduce basic concepts used to derive repair proposals for inconsistent requirements. In Section 4 we introduce the algorithm for calculating minimal and personalized repairs. In Section 5 we discuss our experiences with the developed concepts and indicate topics for future research. Finally, with Section 6 we conclude our paper.

2 Working Example: Digital Camera Recommender

We now introduce a simplified working example from the domain of digital cameras. Fig. 2 depicts a small digital camera assortment consisting of the products $\{p_1, p_2, \ldots, p_8\}$. Those products are stored in the product table P. In our scenario, the customer is interested in cameras supporting the requirements R = {price <= 150, opt-zoom = 5x, sound = yes, waterproof = yes}. These requirements are simply expressed in terms of the query $\sigma_{[R]}P$, where $\sigma_{[R]}$ represents the selection criteria of the query on table P. For example, $\sigma_{[price<=150,\ waterproof=yes]}P$ would result in the recommendation of exactly one product (p_1).

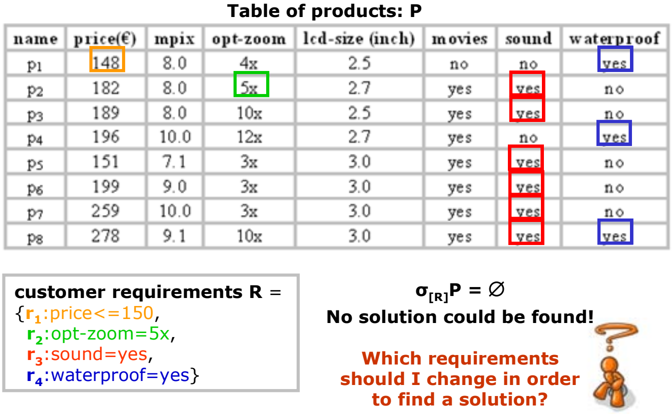


Table of products: P

name	price(€)	mpix	opt-zoom	lcd-size (inch)	movies	sound	waterproof
p1	148	8.0	4x	2.5	no	no	yes
p2	182	8.0	5x	2.7	yes	yes	no
p3	189	8.0	10x	2.5	yes	yes	no
p4	196	10.0	12x	2.7	yes	no	yes
p5	151	7.1	3x	3.0	yes	yes	no
p6	199	9.0	3x	3.0	yes	yes	no
p7	259	10.0	3x	3.0	yes	yes	no
p8	278	9.1	10x	3.0	yes	yes	yes

customer requirements R =
{**r_1**:price<=150,
r_2:opt-zoom=5x,
r_3:sound=yes,
r_4:waterproof=yes}

$\sigma_{[R]}P = \varnothing$
No solution could be found!

Which requirements should I change in order to find a solution?

Fig. 2. Example product assortment (P) and customer requirements (R): R can not be fulfilled – repairs are needed

3 Calculating Minimal Sets of Repair Actions

In our example, querying P with $\sigma_{[R]}$ results in the empty set ($\sigma_{[price<=150,opt-zoom=5x,sound=yes,waterproof=yes]}P=\varnothing$), i.e., there does not exist a product in P that fulfills the given customer requirements. In such a situation, the customer asks herself which requirements should be changed in order to find a solution. Simple product search environments tell the user that no solution could be found. Commercial knowledge-based recommender systems [3][5] propose minimal sets of faulty requirements which should be changed in order to find a solution. However, all existing repair approaches (e.g., [2][13]) focus on identifying and resolving conflicts but do not go further in the sense of *personalizing the selection of repair alternatives*.

The basic idea of existing repair approaches is sketched in Fig. 3. Given a set R={r_1, r_2, …, r_m} of customer requirements and a set P={p_1, p_2, …, p_n} of products, the basic assumption is that the recommender application is able to derive a solution, i.e., $\sigma_{[R]}P$ should be $\neq \emptyset$. However, there are situations where no solution could be found, i.e., $\sigma_{[R]}P = \emptyset$. In such a situation existing knowledge-based approaches calculate a set of diagnoses D = {d_1, d_2, …, d_q}, where $\forall d_i \in D$: $\sigma_{[R - di]}P \neq \emptyset$. The calculation of diagnoses builds on the identification of conflict sets CS = {r_1, r_2, …, r_k}, where CS $\subseteq$ R and $\sigma_{[CS]}P = \emptyset$. A conflict set is said to be minimal if there does not exist a conflict CS' s.t. CS' $\subset$ CS.

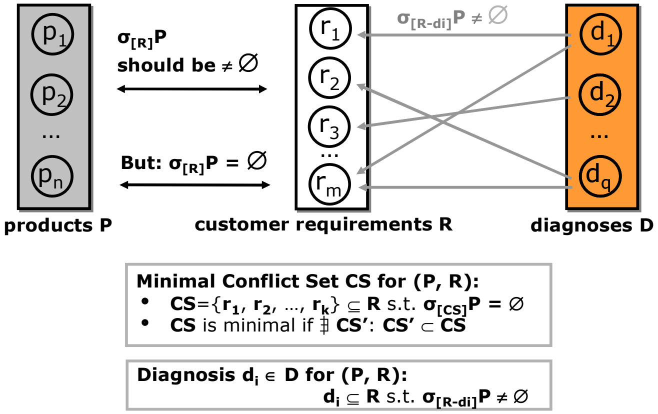

p1
p2
…
pn
σ[R]P
should be ≠ ∅
But: σ[R]P = ∅
r1
r2
r3
…
rm
σ[R-di]P ≠ ∅
d1
d2
…
dq
products P
customer requirements R
diagnoses D
Minimal Conflict Set CS for (P, R):
• CS={r1, r2, …, rk} ⊆ R s.t. σ[CS]P = ∅
• CS is minimal if ∄ CS': CS' ⊂ CS
Diagnosis di ∈ D for (P, R):
di ⊆ R s.t. σ[R-di]P ≠ ∅

Fig. 3. Diagnoses are resolutions of minimal conflicts. Conflicts are calculated by standard algorithms such a [10].

We now exemplify the calculation of diagnoses based on our working example. Given the set of customer requirements R = {price <= 150, opt-zoom = 5x, sound = yes, waterproof = yes}, no solution could be found by the recommender application. The conflicts in R are CS_1:{r_1,r_2}, CS_2:{r_2,r_4}, and CS_3:{r_1,r_3} since $\sigma_{[\{r1,r2\}]}P = \emptyset$, $\sigma_{[\{r2,r4\}]}P = \emptyset$, and $\sigma_{[\{r1,r3\}]}P = \emptyset$.

The calculation of diagnoses is based on the resolution of existing conflicts. Resolution in this context is realized by deleting one of the requirements from the conflict. We can resolve a conflict by deleting exactly one element assuming that the identified conflicts are minimal. Using standard algorithms for the identification of conflicts [10], we can guarantee this minimality property. Having eliminated at least one element from each of the identified conflicts, we are able to present a corresponding diagnosis $d_i \in$ D.

In our working example, the set of identified diagnoses D = {d_1:{r_1,r_2}, d_2:{r_1,r_4}, d_3:{r_2,r_3}} (see Fig. 4). The algorithm for calculating such diagnoses in a personalized way will be discussed in more detail in the next section.

Although we now have identified the set of relevant diagnoses, the user is still left alone with the task of identifying an adequate repair action. Diagnoses only indicate

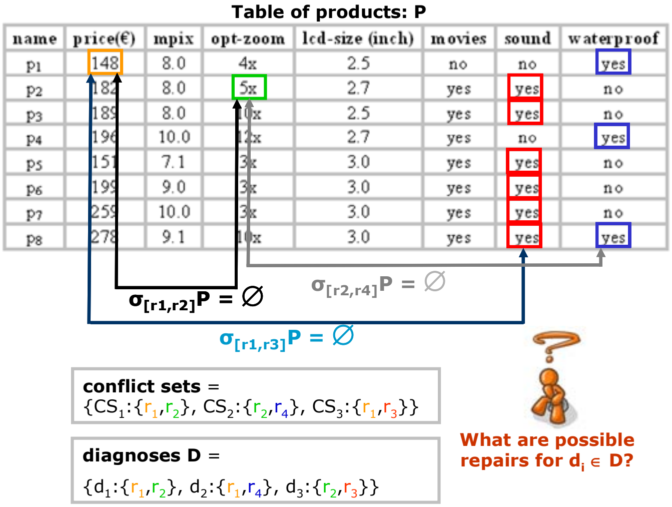


Table of products: P

name	price(€)	mpix	opt-zoom	lcd-size (inch)	movies	sound	waterproof
p1	148	8.0	4x	2.5	no	no	yes
p2	182	8.0	5x	2.7	yes	yes	no
p3	189	8.0	10x	2.5	yes	yes	no
p4	196	10.0	12x	2.7	yes	no	yes
p5	151	7.1	3x	3.0	yes	yes	no
p6	199	9.0	3x	3.0	yes	yes	no
p7	259	10.0	3x	3.0	yes	yes	no
p8	278	9.1	10x	3.0	yes	yes	yes

$\sigma_{[r1,r2]}P = \emptyset$

$\sigma_{[r2,r4]}P = \emptyset$

$\sigma_{[r1,r3]}P = \emptyset$

conflict sets = $\{CS_1{:}\{r_1,r_2\}, CS_2{:}\{r_2,r_4\}, CS_3{:}\{r_1,r_3\}\}$

diagnoses D = $\{d_1{:}\{r_1,r_2\}, d_2{:}\{r_1,r_4\}, d_3{:}\{r_2,r_3\}\}$

Fig. 4. Conflicts between the requirements of R and the products of P. A conflict is resolved if one of its elements is deleted. For example, CS_1 is resolved by deleting r_1.

that certain parameters have to be changed but do not directly indicate the corresponding repair actions. In typical knowledge-based recommender applications, the number of alternative repair actions could potentially be very high which makes the selection of the "best" ones a frustrating task. An example for the calculation of repair actions is depicted in Fig. 5.

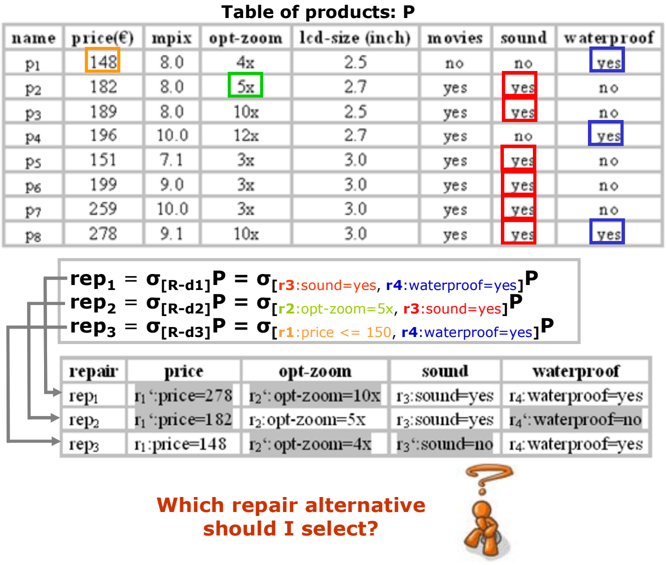


Table of products: P

name	price(€)	mpix	opt-zoom	lcd-size (inch)	movies	sound	waterproof
p1	148	8.0	4x	2.5	no	no	yes
p2	182	8.0	5x	2.7	yes	yes	no
p3	189	8.0	10x	2.5	yes	yes	no
p4	196	10.0	12x	2.7	yes	no	yes
p5	151	7.1	3x	3.0	yes	yes	no
p6	199	9.0	3x	3.0	yes	yes	no
p7	259	10.0	3x	3.0	yes	yes	no
p8	278	9.1	10x	3.0	yes	yes	yes

$\mathbf{rep_1} = \sigma_{[R-d1]}P = \sigma_{[r3:sound=yes,\ r4:waterproof=yes]}P$

$\mathbf{rep_2} = \sigma_{[R-d2]}P = \sigma_{[r2:opt\text{-}zoom=5x,\ r3:sound=yes]}P$

$\mathbf{rep_3} = \sigma_{[R-d3]}P = \sigma_{[r1:price <= 150,\ r4:waterproof=yes]}P$

repair	price	opt-zoom	sound	waterproof
rep1	r1':price=278	r2':opt-zoom=10x	r3:sound=yes	r4:waterproof=yes
rep2	r1':price=182	r2:opt-zoom=5x	r3:sound=yes	r4':waterproof=no
rep3	r1:price=148	r2':opt-zoom=4x	r3':sound=no	r4:waterproof=yes

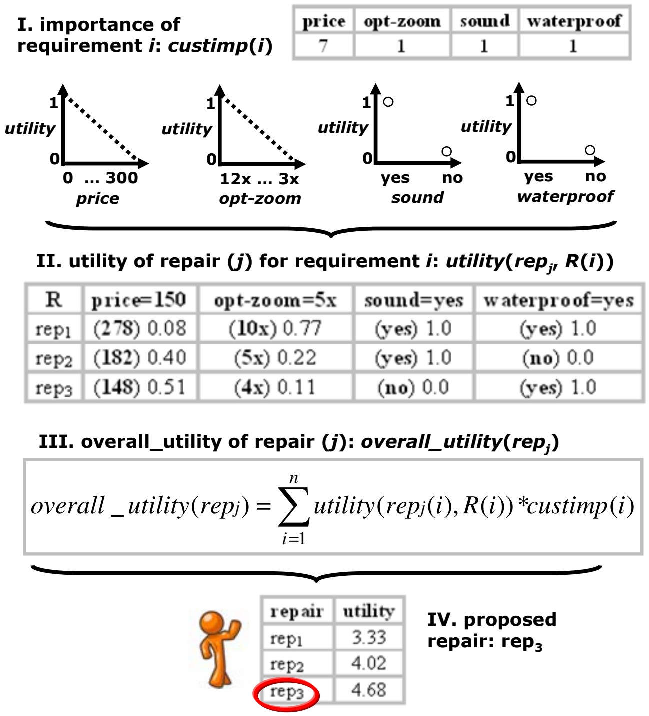
Fig. 5. Simple procedure of calculating repair alternatives

Note that for reasons of simplicity, our example contains only three different repair alternatives.

One repair $rep_j \in$ REP can be calculated on the basis of a diagnosis $d_i \in$ D by querying the product table P as follows: $rep_j \in \sigma_{[R-di]}P$. The set of possible repair actions (REP) is defined as $\cup\sigma_{[R-di]}P$. In our simple digital camera recommender we identify exactly one repair action per diagnosis (see Fig. 5).

4 Utility-Based Calculation of Repair Actions

As already mentioned, the number of repair alternatives can become very high which requires additional support for the user. Our goal is to present those repairs which keep the original requirements the same as much as possible. The derivation of personalized repair actions for a given set of inconsistent customer requirements is sketched in Fig. 6.



I. importance of requirement *i*: *custimp*(*i*)

price	opt-zoom	sound	waterproof
7	1	1	1

II. utility of repair (*j*) for requirement *i*: *utility*(rep_j, *R*(*i*))

R	price=150	opt-zoom=5x	sound=yes	waterproof=yes
rep1	(278) 0.08	(10x) 0.77	(yes) 1.0	(yes) 1.0
rep2	(182) 0.40	(5x) 0.22	(yes) 1.0	(no) 0.0
rep3	(148) 0.51	(4x) 0.11	(no) 0.0	(yes) 1.0

III. overall_utility of repair (*j*): *overall_utility*(rep_j)

$$overall\_utility(rep_j) = \sum_{i=1}^{n} utility(rep_j(i), R(i)) * custimp(i)$$

repair	utility
rep1	3.33
rep2	4.02
rep3	4.68

IV. proposed repair: rep_3

Fig. 6. Determining the most interesting repair alternative

In order to be able to calculate the most interesting repair alternative we need to have additional knowledge about the personal preferences of the customer. This information can derived either by directly asking the customer or by other methods such as analyzing already completed interactions for the existence of preferences. For example, our customer is more interested in the economy of a camera than in the satisfaction of the other requirements. The basic idea behind personalizing repair actions is to identify and present those repair alternatives which potentially have the highest utility for the customer.

Fig. 6 (I.) defines the importance priorities of our example customer – price has definitely the highest importance in this example. The other influence factor for determining the utility of a specific repair alternative is the utility of each individual parameter setting, for example, a low price has a very high utility. In Fig. 6 (II.) the individual utility of each parameter setting of the corresponding repair alternative is shown. For example, the utility of the parameter price of the repair alternative rep_1 is 0.08. By applying the formula (III.), the overall utility of each repair action can be determined. For example, the overall utility of repair action rep_3 is 4.68, which represents the highest utility of the three alternatives. This result is quite intuitive since rep_3 keeps the price as low as possible and takes into account as far as possible the other requirements defined by the customer.

We will now show step-by-step how our algorithm (Algorithm 1) calculates a personalized repair for the given working example (see Fig. 7). The algorithm starts with a call of the conflict detection function which is based on the algorithm of [10]. This function returns one conflict per call. Let us assume, that the first conflict returned is CS_3:{r_1,r_3}. Our algorithm now creates a search tree (HSDAG – Hitting Set Directed Acyclic Graph) with two different branches representing the alternative conflict resolutions. Conflict resolution in the left branch means that the requirement r_1 should not be taken into account, i.e., only products not fulfilling r_1 should be considered further. In the next step, the algorithm evaluates which of the two resolutions creates the remaining product set including the *most interesting remaining alternative* for the customer.

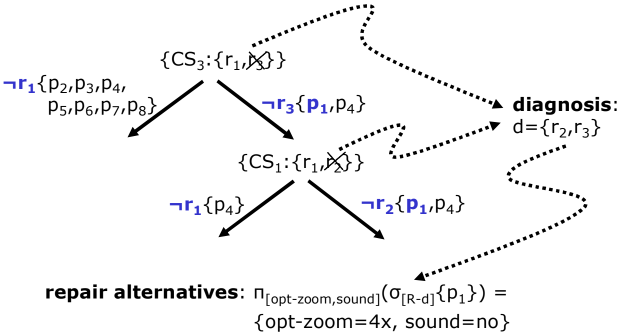

$\{CS_3:\{r_1,\cancel{r_3}\}\}$
$\neg r_1\{p_2,p_3,p_4,p_5,p_6,p_7,p_8\}$
$\neg r_3\{p_1,p_4\}$
diagnosis:
$d=\{r_2,r_3\}$
$\{CS_1:\{r_1,\cancel{r_2}\}\}$
$\neg r_1\{p_4\}$
$\neg r_2\{p_1,p_4\}$
repair alternatives: $\pi_{[opt\text{-}zoom,sound]}(\sigma_{[R\text{-}d]}\{p_1\})$ =
{opt-zoom=4x, sound=no}

Fig. 7. Calculating personalized repair actions: the Customer Requirements Repair Algorithm (Algorithm 1: CREQ-Repairs) applied to our working example

An enlisting of customer-specific product utilities regarding the defined set of requirements is depicted in Table 1. These utilities have been calculated based on the importance distribution defined in Fig. 6. For example, the overall utility of product p_1 (the evaluation is restricted to the specified attributes in R) for our example customer is calculated by 0.51*7+0.11*1+0*1+1*1=4.68.

In the left branch we have to take into account the restriction $\neg r_1$. The product with the highest utility regarding the given set of restrictions $\{\neg r_1\}$ is p_5 (utility 4.5). In the right branch the product with the highest utility (4.68) is p_1 (taking into account the restriction $\neg r_3$). Thus, the algorithm selects the more promising right branch for the following expansions of the search tree. Let us assume that the next conflict returned by the conflict detection algorithm is CS_1:$\{r_1,r_2\}$. The evaluation of the possible tree expansions again results in the selection of the right branch which includes the most promising product p_1. After having deleted r_2 from the set of given requirements, the conflict detection algorithm does not find any further conflicts which means that we have successfully identified one diagnosis $d=\{r_3,r_2\}$ and a corresponding repair proposal {opt-zoom=4x, sound=no}.

The determination of a set of repairs for a given diagnosis is based on the query expression $\pi_{[attributes(d)]}(\sigma_{[R-d]}P)$ where σ denotes the selection criteria and π is the projection onto those attributes contained in the diagnosis d. For example, $\pi_{[opt\text{-}zoom, sound]}(\sigma_{[price<=150,waterproof=yes]}P)$ = {{opt-zoom=4x, sound=no}} which corresponds to the repair alternative rep_3 calculated for the diagnosis in Fig. 7.

Table 1. Utilities of products $\{p_1, \ldots, p_8\}$ for example customer (taking into account the importance values of Fig. 6)

product	price	opt-zoom	sound	waterproof	overall
p_1	3.57	0.11	0	1	**4.68**
p_2	2.8	0.22	1	0	**4.02**
p_3	2.59	0.77	1	0	**4.36**
p_4	2.45	1.0	0	1	**4.45**
p_5	3.5	0.0	1	0	**4.5**
p_6	2.38	0.0	1	0	**3.38**
p_7	0.98	0.0	1	0	**1.98**
p_8	0.56	0.77	1	1	**3.33**

The algorithm for calculating such personalized repair actions is the following (Algorithm 1: CREQ-Repairs). The labeling of the search tree (Hitting Set Directed Acyclic Graph – HSDAG) is based on the labeling of the original HSDAG [12]. A node n in the tree is labeled by a corresponding conflict set CS(n) (see Fig. 7). The set of edge labels from the root to node n is referred to as H(n).

Algorithm 1. CREQ-Repairs ($R\downarrow$,$P\downarrow$,$REP\uparrow$):

(1) Generate a pruned HSDAG T for the collection of conflict sets induced by the requirements in R. The tree is generated in a *best-first* manner, where those nodes are further expanded which include the most interesting remaining product alternatives.

With every theorem prover (TP) call at a node n of T, the satisfiability of $\sigma_{[R-H(n)]}P$ is checked. If no solution could be found, a conflict set CS is returned by the TP, otherwise *ok* is returned.

If $\sigma_{[R-H(n)]}P$ has a solution, a corresponding diagnosis d=H(n) is found.

(2) Return $REP=\pi_{[attributes(d)]}(\sigma_{[R-d]}P)$.

5 Practical Implications

Personalized repair actions for inconsistent customer requirements are an innovative concept which better support customers in their interaction with knowledge-based recommender applications. Although such functionalities do not significantly change the design of the recommender user interface, they clearly contribute to a more intelligent behavior of recommender applications and have the potential to trigger increased trust and satisfaction of users interacting with recommenders.

In real-world knowledge-based recommender applications [3][4][5] we can expect faulty requirements with 2-6 conflicts derived by the conflict detection algorithm. Such conflicts are related to about 50-400 (observations from recommender applications in the domains of financial services and consumer electronics [4]) different repair alternatives which obviously not altogether can be evaluated by the user. Exactly such scenarios were the major motivation for us to think about solutions to personalize the interaction with knowledge-based recommender applications.

The results presented in this paper clearly contribute to improvements in recommender algorithms development. Currently we are further evaluating the presented approach in terms of reduced time efforts (fewer repair alternatives have to be evaluated) and principal product selection behavior. The major question in this context is whether the presented repair alternatives influence the product selection behavior of users.

It is planned to extend the application of the presented personalization concepts to the diagnosis and repair of faulty constraints in recommender knowledge bases. Research in this field has already extensively been conducted (see, e.g., [2]). The next logical step is to develop new and innovative concepts which help domain experts and knowledge engineers to develop and maintain such knowledge bases more effectively. The personalization aspect in this alternative scenario is to identify those sets of faulty constraints which have the highest probability of being responsible for the faulty behavior of the recommender knowledge base (in terms of, for example, faulty products recommended in a certain customer context).

6 Conclusions

In this paper we introduced the concept of utility-based repairs which are extremely important in knowledge-based recommendation scenarios. The presented algorithm shows how to recommend repair actions fitting to the wishes and preferences of the current customer. It is applicable in interactive settings and can be applied to further diagnosis tasks such as the diagnosis of faulty knowledge bases. The contribution of this paper helps developers of knowledge-based applications to better support their users in the interaction with the system and promises significant time savings in the interaction phase.

Acknowledgements

The results presented in this paper have been developed within the scope of the WE-CARE project (Austrian Research Agency, Grant Nr. 820128).

References

1. Burke, R.: Knowledge-based Recommender Systems. Encyclopedia of Library and Information Systems 69(32) (2000)
2. Felfernig, A., Friedrich, G., Jannach, D., Stumptner, M.: Consistency-based Diagnosis of configuration knowledge bases. AI Journal 152(2), 213–234 (2004)
3. Felfernig, A., Kiener, A.: Knowledge-based Interactive Selling of Financial Services with FSAdvisor. In: 17th Innovative Applications of Artificial Intelligence Conference (IAAI 2005), pp. 1475–1482. AAAI Press, Pittsburgh (2005)
4. Felfernig, A., Friedrich, G., Jannach, D., Zanker, M.: An Environment for the Development of Knowledge-based Recommender Applications. International Journal of Electronic Commerce (IJEC) 11(2), 11–34 (2006)
5. Felfernig, A., Isak, K., Szabo, K., Zachar, P.T.: The VITA Financial Services Sales Support Environment. In: AAAI/IAAI 2007, Vancouver, Canada, pp. 1692–1699 (2007)
6. Felfernig, A.: Standardized Configuration Knowledge Representations as Technological Foundation for Mass Customization. IEEE Transactions on Engineering Management 54(1), 41–56 (2007)
7. Felfernig, A., Burke, R.: Constraint-based Recommender Systems: Technologies and Research Issues. In: Proceedings of the ACM International Conference on Electronic Commerce, Innsbruck, Austria, August 19-22, pp. 17–26 (2008)
8. Greiner, R., Smith, B., Wilkerson, R.: A correction to the algorithm in Reiter's theory of diagnosis. Artificial Intelligence 41(1), 79–88 (1989)
9. Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: GroupLens: applying collaborative filtering to Usenet news Full text. Communications of the ACM 40(3), 77–87 (1997)
10. Junker, U.Q.: QuickXPlain: Preferred Explanations and Relaxations for Over-Constrained Problems. In: AAAI 2004, pp. 167–172. AAAI Press, San Jose (2004)
11. Pazzani, M., Billsus, D.: Learning and Revising User Profiles: The Identification of Interesting Web Sites. Machine Learning (27), 313–331 (1997)
12. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence 23(1), 57–95 (1987)
13. O'Sullivan, B., Papadopoulos, A., Faltings, B., Pu, P.: Representative Explanations for Over-Constrained Problems. In: 22nd National Conference on AI (AAAI 2007), pp. 323–328 (2007)
14. Wilson, D., Martinez, T.: Improved Heterogeneous Distance Functions. Journal of Artificial Intelligence Research 6, 1–34 (1997)

An Ecological Model-Based Reasoning Model to Support Nature Park Managers

Mark Hoogendoorn, Jan Treur, and Muhammad Umair

Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081, 1081 HV Amsterdam
{mhoogen,treur,mumair}@few.vu.nl
http://www.cs.vu.nl/~{mhoogen,treur,mumair}

Abstract. A decision support system model is described to advise nature park managers. It applies dynamic modelling techniques to relate abiotic characteristics of a site over time to species that can be found. For a desired vegetation type it determines how such abiotic factors are to be changed in order to obtain the desired change in the vegetation after some time. The system takes into account an ecological model of the temporal dynamics of species, interspecies interactions, and abiotic factors. Applying model-based reasoning and dynamical systems methods to this ecological model, decision options on abiotic conditions are determined in order to obtain desired vegetation types.

1 Introduction

Plants only grow in areas with suitable (abiotic) species-specific environmental conditions. When such abiotic conditions change over time, as a consequence the vegetation of a site may also change. How the occurrence of a species relates to a terrain's abiotic (physical and chemical) characteristics can be expressed as environmental preferences of a species. More specifically, the abiotic preferences of species for factors such as acidity, nutrient value and moisture, are decisive for the question whether or not they can become part of the vegetation on a specific site. The appreciation of a nature park usually lies in the type of flora (and fauna) that can be found. In contrast, measures that can be taken by a manager usually concern the abiotic factors, such as the ground water level. Therefore knowledge about abiotic preferences of plant species, are a crucial element to be used by nature managers in their management. However, nature managers responsible for terrains do not always possess such detailed knowledge, and in particular are not fully aware of how the dynamics of the relationships between abiotic factors and occurrence of species work out over time. Several models have been proposed to represent such relations in a mathematical or logical way (see e.g. [4-9]).This paper describes a decision-support system model that has been designed to support them in their decision making processes. Once an analysis of a terrain has been made, taking into account both the dynamics of abiotic and biotic factors and their interaction, nature managers can use this to manage the terrain, for example by taking measures on abiotic factors to improve the vegetation quality of the site.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 172–182, 2009.

The decision-support system model was designed on the basis of temporal dynamical modelling of the dynamics of the abiotic and biotic factors and their relations, and temporal model-based reasoning techniques using the dynamic model. Both a quantitative modelling as well as a qualitative approach are taken, and are applied to a case study within the specified domain. The approaches allow a manager of a nature park to set certain (long term) goals, and can derive using what settings are appropriate to reach these goals.

This paper is organised as follows. In Section 2 the domain of application is treated in more detail. The quantitative and qualitative models are explained in Section 3. Section 4 presents a model-based reasoning approach in order to enable reasoning about the qualitative model (e.g. determine how certain long term ecological gaols can be reached). A mathematical approach to achieve the same is presented in Section 5, and finally, Section 6 concludes the paper.

2 Domain of Application

The domain considered is that of a nature park manager who has to make decisions about his or her terrain. Within such decision processes both abiotic factors and biotic factors play an important role.

Abiotic preferences of a species. Every plant species needs a combination of abiotic conditions to grow at a given site: its *abiotic preferences*. For example, the abiotic preferences of *Caltha palustris* L., are: very moist or fairly wet; basic, neutral or slightly acid; nutrient poor, fairly nutrient rich or nutrient rich terrain. For the species *Poa trivialis* L. a terrain needs to be fairly moist, very moist or fairly wet; basic or neutral; nutrient rich or very nutrient rich.

Biotic preferences of a species. For a simple approximation the abiotic preferences of a species can be used to determine whether or not a species can grow. However, also interaction between species can play a role, in the sense that the presence of another species may affect a given species in a positive or negative manner: in addition to abiotic preferences, also such *biotic preferences* of species can be used. Some well-known examples of such interactions between two species are:

interaction	effect of species 2 on species 1	effect of species 1 on species 2
competition	negative	negative
symbiosis	positive	positive
parasitism	positive	negative

Such biotic preferences imply that the suitability of a site does not only depend on the abiotic characteristics of the site and abiotic preferences of species. Dynamic patterns over time may result, such as the periodic predator-prey cyclus as known from the literature. Dynamic modelling methods are required to address such patterns.

Ecological decision making. A manager of a nature park has the possibility to manipulate certain factors on the site. Often such factors concern the abiotic circumstances, such as the (ground) water level. However, also the introduction of certain species such as grazers may be included in the set of instruments available to the park manager. This indicates that, in line with what was discussed above, both the dynamics of interactions between species and abiotic factors and the dynamics between different species are to be taken into account within ecological decision making processes. This is the challenge addressed in subsequent sections.

3 An Example Ecological Model

In this section an example ecological model is presented both in qualitative and quantitative format. The model is presented in a simplified form with two competitive species s_1 and s_2 which both depend on the abiotic factor moisture. It is only used for the purpose of illustration. The method presented can be applied to any ecological model. Figure 1 shows a causal diagram for this model. Below, both a qualitative and a quantitative variant of the model are introduced.

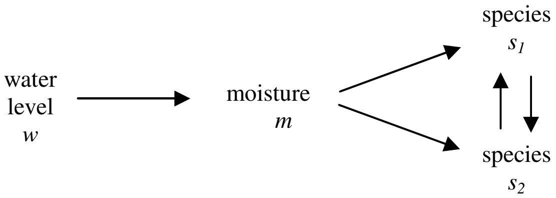


Fig. 1. Causal diagram of ecological model

3.1 Quantitative Model

A differential equation form for this model is as follows:

$$\begin{aligned} d\,s_1(t)\,/\,dt &= \beta\ {*}s_1(t) * [c(t) - a_1{*}s_1(t) - a_2{*}s_2(t)] \\ d\,s_2(t)\,/\,dt &= \gamma{*}s_2(t) * [c(t) - b_1{*}s_1(t) - b_2{*}s_2(t)] \\ d\,c(t)/dt &= \omega{*}(\eta\, m(t)\ -\ c(t)) \\ d\,m(t)\,/dt &= \Theta * (\lambda\, w - m(t)) \end{aligned}$$

Here $s_1(t)$ and $s_2(t)$ are the densities of species s_1 and s_2 at time point t; moreover, $c(t)$ denotes the carrying capacity for s_1 and s_2 at t, which depends on the moisture $m(t)$. The moisture depends on the water level indicated by w. This w is considered a parameter that can be controlled by the manager of the terrain, and is kept constant over longer time periods. Moreover the parameters β, γ are growth rates for species s_1, s_2. For carrying capacity and moisture respectively, η and λ are norm proportion parameters, and Θ and ω are speed factors. The parameters a1, a2 and b1, b2 are proportional contribution in the competitive environment for species s1 and s2 respectively. Based on the quantitative model discussed a large number of simulations have been performed resulting in a variety of interesting patterns. An example situation is where the proposed decision support system assists the nature park manager to estimate the densities of species s_1 and s_2 after a certain time (e.g. 10 years) given the current abiotic circumstances (moisture m) that depends on water level w. Figure 2 shows the results for this particular situation.

Note that for this example set of equations, equilibria can be determined as follows:

$$\beta * s_1 * [c - a_1*s_1 - a_2*s_2] = 0 \qquad \gamma * s_2 * [c - b_1*s_1 - b_2*s_2] = 0$$
$$\eta m - c = 0 \qquad \lambda w - m = 0$$

This can be solved by

$$m = \lambda w \quad \text{and} \quad c = \eta m = \eta\lambda w$$

Moreover,

$$s_1 = 0 \quad \text{or} \quad c - a_1*s_1 - a_2*s_2 = 0 \quad \text{and} \quad s_2 = 0 \quad \text{or} \quad c - b_1*s_1 - b_2*s_2 = 0$$

This is equivalent to

Either $s_1 = s_2 = 0$
or $s_1 = c/a_1$ and $s_2 = 0$
or $s_1 = 0$ and $s_2 = c/b_2$
or $s_1 = c\,(a_2 - b_2))/(\,a_2\,b_1 - a_1\,b_2)$ and $s_2 = -\,c\,(a_1 - b_1))/(\,a_2\,b_1 - a_1\,b_2)$

Note that in this simple example model, the equilibria can be determined analytically. However, for the general case it is not assumed that equilibria can be determined in an analytic manner, and it is not assumed that the terrain will reach an equilibrium state within the time period considered.

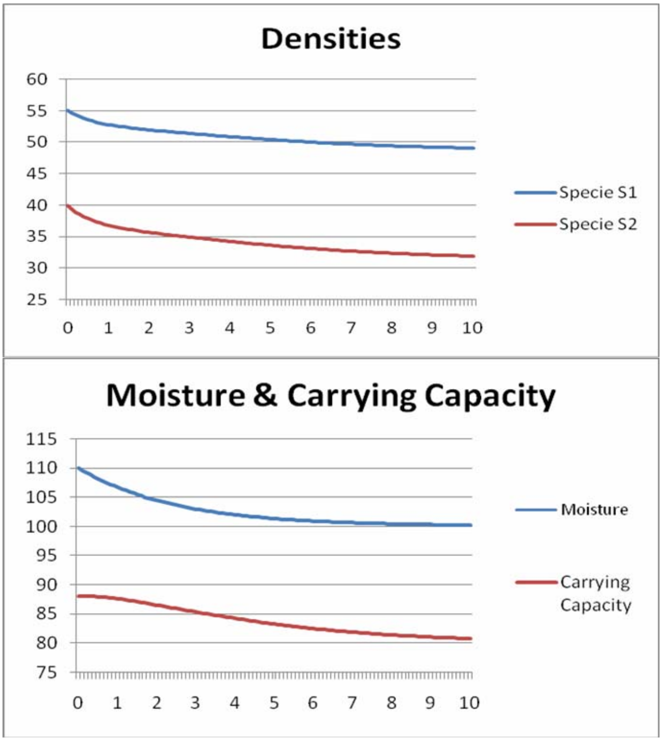


Fig. 2. Predicted densities, moisture and carrying capacity after 10 years (w=200, m(0)=110, c(0)=88, λ=0.5, η=0.8, β=0.01, γ=0.02, Θ=0.4, ω=0.4)

3.2 Qualitative Model

Below, a model is introduced whereby a qualitative perspective is taken. Hereby, a number of dependency rules are specified that indicate how the abiotic world and species interact. These dependencies are represented by means of so called leadsto_after rules (cf. [1]). The format of the leadsto_after rule is as follows:

leadsto_after: INFO_ELEMENT x INFO_ELEMENT x DURATION

indicating that from the first information element the second can be derived, whereby the second element becomes true after duration d. The truth of information elements over time is indicated by the at predicate:

at: INFO_ELEMENT x TIME

Which specifies that the information element is true at the given time point. Given these constructs, properties of states can easily be derived. A simple forward reasoning rule is the following:

leadsto_after(I1, I2, D) ∧ at(I1, T) → at(I2, T+D)

More specifics about such reasoning rules are given in Section 4. Below, two variants of a model are specified. One whereby the density of a species depends solely on the abiotic factors, and one which allows for interaction with other species.

Non-interacting species model. A simple model whereby the densities of species merely depend upon the moisture level (which in turn is dependent upon the water level) can be represented using the following leadsto_after rules:

leadsto_after(waterlevel(X), moisture(X), 1) where X can e.g. be low, medium or high.

Furthermore, for the dependency between abiotic factors and the density of the population, the following (generic) relationship can be identified:

```
leadsto_after(and(X, abiotic_preference_for(S, X, pos), has_density(S, D1),
                  next_higher_density(D1, D2)), has_density(S, D2), 1)

leadsto_after(and(X, abiotic_preference_for(S, X, neg), has_density(S, D1),
                  next_higher_density(D2, D1)), has_density(S, D2), 1)
```

Hereby, the preferences can be made explicit, for instance the fact that a particular species prefers high moisture: abiotic_preference_for(species1, moisture(high), pos).

Interacting species model. Not having any influence of the different species residing at a particular location is not very realistic. Therefore, the model can be extended to incorporate this factor as well. Hereby, a larger number of leadsto_after rules are required as more complex interactions occur. Two example rules are shown below.

```
leadsto_after(and(X, abiotic_preference_for(S1, X, pos),
                  abiotic_preference_for(S2, X, pos), biotic_preference_for(S1, S2, pos),
                   has_density(S1, D1), next_higher_density(D1, D2)),
                   has_density(S1, D2), 1)
leadsto_after(and(X, abiotic_preference_for(S1, X, pos),
                 abiotic_preference_for(S2, X, neg), biotic_preference_for(S1, S2, neg),
                 has_density(S1, D1), next_higher_density(D1, D2)),
                 has_density(S1, D2), 1)
```

Hereby, the abiotic interactions can be set in the same way as specified before, whereas the biotic preferences are explicitly represented, for instance a competition between two species 1 and 2:

biotic_preference_for(species1, species2, neg) biotic_preference_for(species2, species1, neg)

parasitism between species 1 and 2:

biotic_preference_for(species1, species2, pos) biotic_preference_for(species2, species1, neg)

or symbiosis between species 1 and 2:

biotic_preference_for(species1, species2, pos) biotic_preference_for(species2, species1, pos)

4 Decision Support by Model-Based Temporal Reasoning

This section shows how model-based temporal reasoning can be utilized to support nature park managers in reaching the goals they want to set for a certain nature region. The model based-reasoning approach is therefore presented first, after which examples are shown using the qualitative modelling approach. The rules within the reasoning mechanism are specified in an executable logical format called LEADSTO [2]. The basic building blocks of this language are temporal causal relations denoted by $\alpha \twoheadrightarrow_{e, f, g, h} \beta$, which means:

if state property α holds for a certain time interval with duration g,
then after some delay (between e and f) state property β
will hold for a certain time interval of length h.

where α and β are state properties of the form 'conjunction of literals' (where a literal is an atom or the negation of an atom), and e, f, g, h non-negative real numbers. The LEADSTO language features a simulation engine; for more details, see [2].

For both temporal forward and backward simulation well-known reasoning techniques can be applied. An example of a temporal forward reasoning rule is shown below. Hereby a focusing mechanism is used as well, indicating what information elements to focus on. How this focusing mechanism is used is stated in the example case. Note that the subscript below the LEADSTO arrow has been left out, meaning that the standard parameters 0,0,1,1 are used.

P1: Positive forward simulation

If I holds at T and it is known that I leads to J after duration D, and J is in focus, then the J holds after D.

$\forall$I,J:INFO_ELEMENT $\forall$D:DURATION $\forall$T:TIME
at(I, T) $\wedge$ leads_to_after(I, J, D) $\wedge$ in_focus(J) $\twoheadrightarrow$ at(J, T+D)

More forward reasoning rules exist, see [1] for more details. For backward reasoning the abduction principle can be applied:

P2: Positive backward simulation

If it is believed that J holds at T and that I leads to J after duration D, and I is in focus, then it is believed that I holds before duration D.

$\forall$I,J:INFO_ELEMENT $\forall$D:DURATION $\forall$T:TIME
at(J, T) $\wedge$ leads_to_after(I, J, D) $\wedge$ in_focus(I) $\twoheadrightarrow$ at(I, T-D)

The results of applying this rule are not guaranteed to be correct since there could be multiple leads_to_after rules that cause J to occur. Again, see [1] for more details and backward simulation rules.

In the example case, two species of plants are considered s_1 and s_2. The density of the species can have three values: low, medium, and high; for the water level and moisture the same values are allowed. The specific interactions are listed in Table 1.

Table 1. Density change conditions

Preference of s1for abiotic factor	**Preference of s2 for abiotic factor**	**Interaction between s1 and s2**	**Growth of density of s1**
pos	pos	pos	+1
pos	pos	neg	0
pos	neg	pos	0
pos	neg	neg	+1
neg	pos	pos	0
neg	pos	neg	-1
neg	neg	pos	-1
neg	neg	neg	0

For instance, when looking at the first column in the table, in case s_1 has a preference for the current abiotic circumstances, and so does s_2, and they have a positive interaction, then the density of s_1 grows by one as well (e.g. from medium to high). Of course, in case the highest value is reached, the growth no longer occurs. Below, two specific cases are addressed, one for symbiosis between s_1 and s_2, one for a competitive relationship. Note that it is assumed that at least a low population size of each species is present. If also no plants of a species would be taken into account the interaction between the species would become dependent upon the presence of these species. In the quantitative model this information is taken into account.

Symbiosis. The first case considered is symbiosis. Hence, the two plants have a positive influence upon each other. Furthermore, both plants prefer medium or high moisture, and dislike low moisture. The manager of the park want to find out what would happen in case he decides to lower the water level in the park. Using forward simulation the proposed support system starts reasoning (given the initial conditions that the current density of both plants is high, and the water level will be low during the coming 5 years). Figure 3 shows the results. Hereby, the left part of the figure denotes the atoms that occur during the simulation run, whereas the right side indicates the simulation time line where a dark box indicates the atom is true at that time point, and a light box indicates false. Note that the arguments in the atoms specify the real world time points derived, which do not have any relationship with the simulation time.

It can be seen in the trace that at simulation time 1 the predicted moisture levels are calculated, all being low as well:

at(moisture(low, 1) at(moisture(low, 2) at(moisture(low, 3) at(moisture(low, 4)

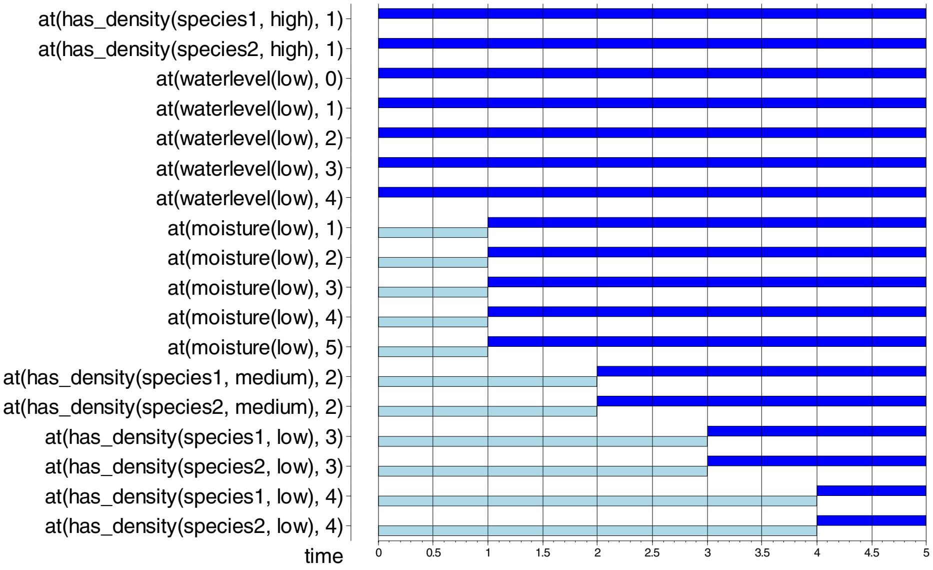


Fig. 3. Forward reasoning for symbiosis case

Furthermore, the densities are calculated. Due to both species disliking the low moisture level, both population densities are predicted to decrease to a medium density within 2 years:

at(has_density(species1, medium), 1) at(has_density(species2, medium), 1)

After that, the densities will even increase to a low level.

Competition. In the second case, there are two competitive species, whereby s_2 prefers low moisture, and does not prefer other moisture types, whereas s_1 prefers non-low moisture types. The manager of the nature park wants to establish a high level of

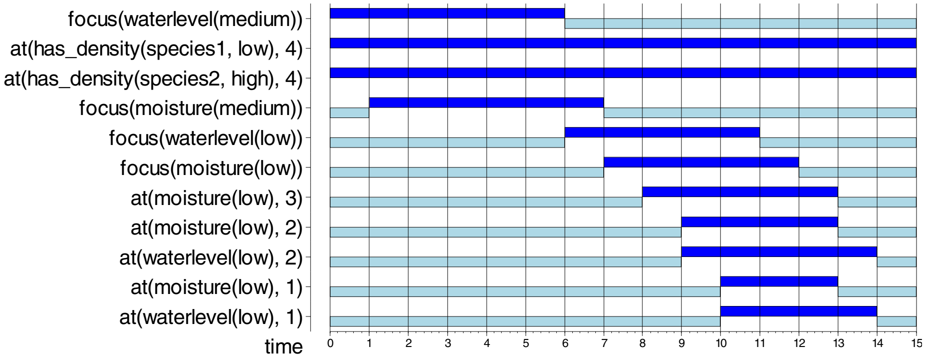


Fig. 4. Backward reasoning for competitive case

s_2, and a low level of s_1, to be established after four years. The manager can ask for advice from the support system, and also has to set a focusing mechanism since the backward reasoning can deliver a lot of results. Therefore, the manager sets a preference for a medium water level, after which a low water level should be considered. In Figure 4 the resulting trace is shown.

It can be seen that the initial goals of the manager are inputted into the system:

at(has_density(species1, low), 4) at(has_density(species2, high), 4)

Thereafter the reasoning starts (of which only the moisture and water levels are shown for the sake of brevity). The focus is initially set to a medium water level to establish the overall goal: focus(waterlevel(medium)). The backward reason does however not result in possible solutions, therefore the focus is set to a low water level. Here, a solution is indeed found, namely to immediately set the water level to low next year, resulting in the goal being reached.

5 Decision Support by Parameterised Temporal Projection

Differential equations for the sensitivities of values of the variables w.r.t. the parameter *w* are obtained by differentiating the original differential equations for *w*:

$$\begin{aligned}
\partial\, \partial s_1/\partial w/\partial t &= \beta *\partial s_1(t)/\partial w\ [c(t) - a1*s_1(t) - a2*s_2(t)] + \beta *s_1(t) * [\partial c(t)/\partial w - \\
&\qquad a1*\partial s_1(t)/\partial w - a2*\partial s_2(t)/\partial w] \\
\partial\, \partial s_2/\partial w /\partial t &= \gamma *\partial s_2(t)/\partial w\ [c(t) - b1*s_1(t) - b2*s_2(t)] + \gamma *s_2(t) * [\partial c(t)/\partial w - \\
&\qquad b1*\partial s_1(t)/\partial w - b2*\partial s_2(t)/\partial w] \\
\partial\, \partial c/\partial w /\partial t &= (\ \eta\ \partial m(t)/\partial w\ - \partial c(t)/\partial w) * \omega \\
\partial\, \partial m/\partial w /\partial t &= (\lambda - \partial m(t)/\partial w) * \Theta
\end{aligned}$$

These equations describe how the values of species s_1, s_2, moisture m and carrying capacity c at time point t are sensitive to the change in the value of the water level parameter w. Figure 2 shows the trend in change of densities of species over 10 years given the initial values of abiotic circumstance (water level w). Using the following formula, the nature park manager can determine the change (Δw) in abiotic circumstance w to achieve the goal at some specific time point in future.

$$\Delta w = [s_1(w+\Delta w) - s_1(w)] / (\partial s_1/\partial w)$$

where $s_1(w+\Delta w)$ is the desired density at time t, $s_1(w)$ the predicted density s_1 at time t for water level w, and $(\partial s_1/\partial w)$ the change in density of s_1 at time t against the change in w. Figure 2 depicts a situation where the densities of species s_1 and s_2 are predicted to decrease, given $w = 200$. Under these settings the density of species s_1 will be *49.* If the nature park manager wants to aim it to become *55* after 10 years, then according to the model described above he or she has to change w to *240* (see Figure 5).

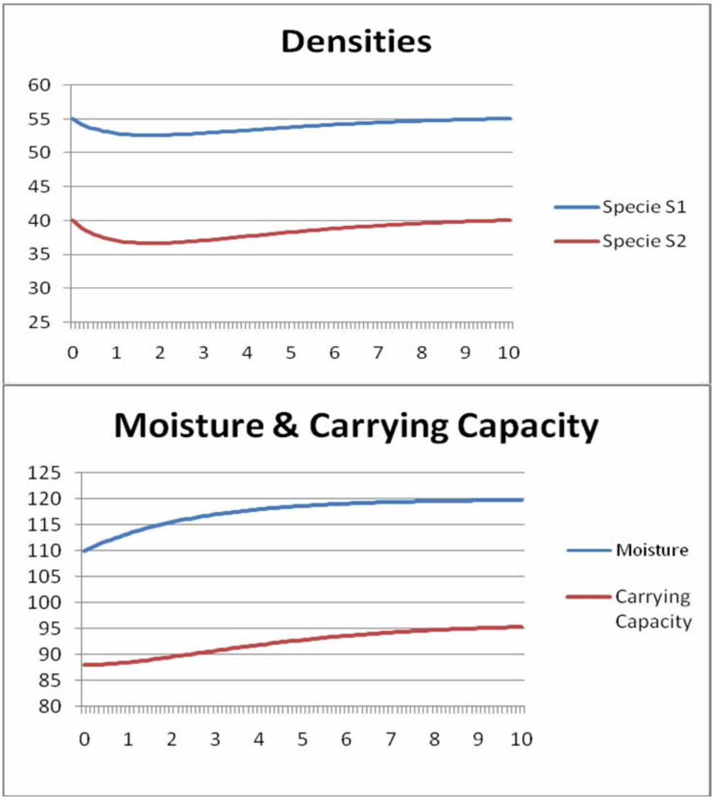


Fig. 5. Densities, moisture and carrying capacity over 10 years, after incorporating Δ*w*= *40* (w=240, m(0)=110, c(0)=88, λ=0.5, η=0.8, β=0.01, γ=0.02, Θ=0.4, ω=0.4)

6 Conclusions and Related Work

In this paper, an approach has been presented to support managers of nature parks. In order to make this support possible, both a quantitative and qualitative approach have been presented. For both types of models, approaches have been proposed to simulate populations over time as well as setting a certain goal to be reached within a certain period, and deriving how the circumstances can be adjusted to achieve these goals. A qualitative approach has advantages in possibilities for explanation. However, in contrast to qualitative approaches, quantitative approaches cover also cases where only small gradual changes occur that accumulate over time into larger differences. Both approaches have been extensively evaluated using a dedicated case study.

During the last decade, decision support for nature park managers is an area that is addressed more and more, often in combination with GIS-systems; see, for example, [6]. Some of the more known systems proposed are SELES [4] and EMDS [5;7]. Both approaches allow to take into account spatial aspects, which are not addressed yet in the current paper. SELES subsumes aspects of cellular automata, discrete event simulation, and Markov chains, and is primarily based on stochastic Monte Carlo simulations, with fewer possibilities for deterministic models. A difference of the approach put forward in the current paper is that deterministic models are addressed and that both qualitative and quantitative models are supported. EMDS is a mainly qualitative approach and has fewer possibilities to take into account the dynamics of processes over longer time periods. The decision component of EMDS was worked out based on hierarchical multi-criteria decision making. A difference of the approach put forward in the current paper is the emphasis on modelling the dynamics of the processes over time, and the possibility for both qualitative and quantitative models.

Other qualitative models have been proposed as well. Salles *et al.* [9], for example introduce a qualitative model for two interacting species. They form a qualitative theory of these interactions, and implement them within the GARP system [3]. Using this system, a variety of interactions can be simulated using forward simulation. They do however not address the possibility to perform backward reasoning such as addressed in this paper. In [8] another qualitative approach is proposed, modelling more complex interactions between populations in a so-called ant's garden setting. Again, reasoning backward in order to determine how a certain desired state can be reached is not addressed. The authors attempt to run a number of simulations to see whether they result in the appropriate end result.

References

1. Bosse, T., Both, F., Gerritsen, C., Hoogendoorn, M., Treur, J.: Model-Based Reasoning Methods within an Ambient Intelligent Agent Model. In: Mühlhäuser, M., Ferscha, A., Aitenbichler, E. (eds.) Constructing Ambient Intelligence: AmI 2007 Workshops Proceedings. Communications in Computer and Information Science (CCIS), vol. 11, pp. 352–370. Springer, Heidelberg (2008)
2. Bosse, T., Jonker, C.M., van der Meij, L., Treur, J.: A Language and Environment for Analysis of Dynamics by Simulation. International Journal of Artificial Intelligence Tools 16, 435–464 (2007)
3. Bredeweg, B.: Expertise in Qualitative Prediction of Behaviour. PhD thesis. University of Amsterdam, Amsterdam, The Netherlands (1992)
4. Fall, A., Fall, J.: A Domain-Specific Language for Models of Landscape Dynamics. Ecological Modelling 141, 1–18 (2001); Earlier version: Fall,J., Fall, A.: SELES: A spatially Explicit Landscape Event Simulator. In: Proc. of the Third International Conference on Integrating GIS and Environmental Modeling, Sante Fe (1996)
5. Gärtner, S., Reynolds, K.M., Hessburg, P.F., Hummel, S., Twery, M.: Decision support for evaluating landscape departure and prioritizing forest management activities in a changing environment. Forest Ecology and Management 256, 1666–1676 (2008)
6. Rauscher, H.M., Potter, W.D.: Decision support for ecosystem management and ecological assessments. In: Jensen, M.E., Bourgeron, P.S. (eds.) A guidebook for integrated ecological assessments, pp. 162–183. Springer, New York (2001)
7. Reynolds, K.M.: Integrated decision support for sustainable forest management in the United States: fact or fiction? Computers and Electronics in Agriculture 49, 6–23 (2005)
8. Salles, P., Bredeweg, B., Bensusan, N.: The Ants Garden: Qualitative models of complex interactions between populations. Ecological Modelling 194(1-3), 90–101 (2006)
9. Salles, P., Bredeweg, B.: Modelling Population and Community Dynamics with Qualitative Reasoning. Ecological Modelling 195(1-2), 114–128 (2006)

Calculating Decoy Items in Utility-Based Recommendation

Erich Christian Teppan and Alexander Felfernig

University of Klagenfurt
Universitätsstr. 65-67, 9020 Klagenfurt, Austria
{erich.teppan,alexander.felfernig}@uni-klu.ac.at

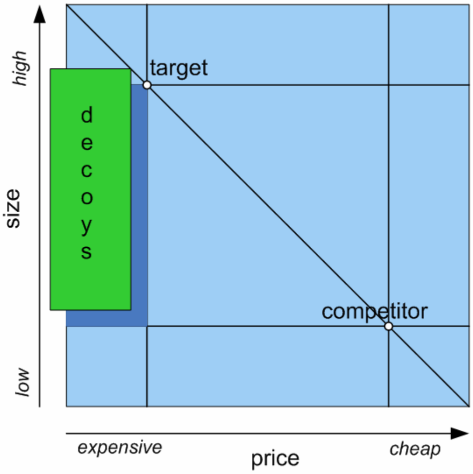
Abstract. Recommender systems support internet users in the often awkward task of finding suitable products in a vast and/or complex product assortment. Many different types of recommenders have been developed during the last decade. From a technical point of view those approaches already work well. What has been widely neglected are decision theoretical phenomenons which can severely impact on the optimally of the taken decision as well as on the challenge to take a decision at all. This paper deals with decoy effects, which have already shown big persuasive potential in marketing and related fields. The big question to be answered in this paper is how to automatically calculate decoy effects in order to identify unforeseen side effects. This includes the presentation of a new decoy model, its combination with utility values calculated by a recommender system, an empirical evaluation of the model, and a corresponding user interface, which serves as starting point for controlling and implementing decoy effects in recommender systems.

1 Introduction

Because of the complexity and size of some presented product assortments on various internet platforms, potential customers often find it hard to identify products which match their wishes and needs. Various approaches of recommender systems [2][4][5] have been developed in order to target this challenge. *Collaborative filtering* [2][4][5] recommenders are typically found on sites from vendors of vast but uncomplex products like books. Basing on information of previous purchases and ratings such systems try to find similar users and present items which have been rated positively by those neighbor users but haven't already be presented to the user itself ('People that liked X also liked Y'). *Content-based recommendation* [2][4][5] exploits information about the items themselves, for example which genre, main actor, etc. in the domain of movies. By maintaining a user profile which describes what the user is interested in (often expressed by keywords) the system is able to find items the user probably is interested in. Those approaches, which can be well combined [2][4][5], lack some problems. Apart from the cold start problem of collaborative systems the content-based as well the collaborative approach are not able to describe complex product domains like for example legislative restrictions of financial products or technical affordances and compatibility issues of high tech products. These are areas where k*nowledge-based recommenders* (KBRs) [2][4][5] can be applied. Typically

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 183–192, 2009.

KBRs offer some kind of user dialog where the user can specify her/his requirements. Contradicting user input can be recognized by the system and furthermore KBRs offer recommendations how to correct such inconsistent user input [5]. The domain knowledge and rules are stored in a knowledge base. This makes it possible to give the user additional information about how/why to change requirements or why a certain product is matching the user requirements [5]. Very common for KBRs is the combination with *utility-based approaches* [2]. Utility-based approaches like *multi-attribute-utility-theory* (MAUT) [11] serve the aim of calculating a utility for all items for a specific user and thus are able to order the items on the result pages. One important aspect which comes along with the presentation of products, although ordered, is that decision phenomenons often lead to unforeseen side effects concerning the perception of products [9]. This means that the perceived strengths and weaknesses of a specific item can change in the light of the presented set of items. If the presented items are competitive this is evident. Interestingly also irrelevant items (i.e. items which are clearly inferior) have a strong impact on the decision task [7]. This class of phenomenons is known as *decoy effects* [8]. One of the most well-known decoy effects is the *asymmetric dominance effect* (ADE) [1][10]. An ADE is occurring when to a set of competitive items a fully dominated item (i.e. inferior in all attributes) is added. In such a case the selection distribution changes such that the dominating item(s) (i.e. the item which is dominating the inferior item) are selected more often. A dominated item is also called *decoy*, a dominating item is called *target*, and a non-dominating and non-dominated item is called *competitor*.



without decoy	T		C
Size	21"		19"
Price	400 €		200 €

with decoy	T	D	C
Size	21"	20,5"	19"
Price	400 €	410 €	200 €

Fig. 1. Example of asymmetric dominance effect and relative positions of items

Fig. 1 is showing a simple example of an asymmetric dominance effect in the TV domain and the relative positions of possible decoy items: Without addition of a decoy item (*D*) the set only consists of two non-dominated TVs. The target (*T*) is larger and the competitor is cheaper. After adding *D*, the attractiveness of *T* increases as *T* totally dominates *D* (i.e. is better in all dimensions) whereas *C* does not (i.e. is still worse in the dimension *size*). Such constellations lead to a higher probability of selection for target items. The ADE can be seen as the special case of two further decoy effects: the *attraction effect* and the *compromise effect* [8]. Fig. 2 (a) is summarizing the relative positions of the three effects in the two-dimensional case. On a recommender result page things are more complicated. Typically there are multiple items. This produces often highly complex interactions.

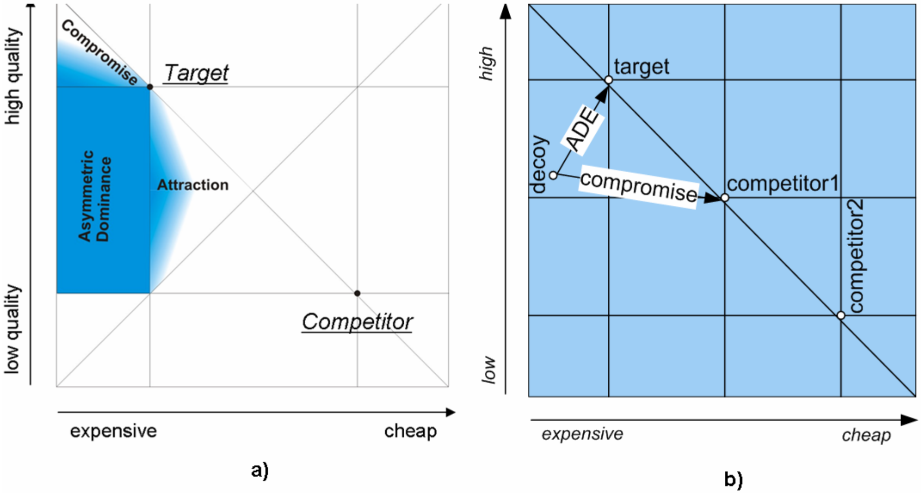


Fig. 2. Decoy effects and the corresponding relative item positions

Fig. 2 (b) is showing an example which consists of only one more competitive item (competitor1, competitor2, target). Introducing an asymmetric dominated decoy for the target also results in at least one more decoy effect: In this case the decoy also acts as a *compromise decoy* for competitor1. One of the factors which is additionally influencing the strength of a decoy, is the similarity (i.e. relative distance) of the items [3]. When there are more than two attribute dimensions the interactions are getting even more complicated. On a recommender result page it is very unlikely that there occur no decoy effects. In order to avoid any form of decoy effect, it would be necessary that all presented items are perfectly competitive, i.e. no dominated alternatives and no tradeoff contrasts [1][9]. In other words, every item has the same overall utility, which is typically not the case. For these reasons a tool is needed which is able to calculate choice-set dependent dominance values of presented items, and thus is able to detect unforeseen side effects. The detection of such effects offers the possibility of counter attacking, for example by informing the user about such constellations when presenting the result page. One further advantage of such a system is the possibility of calculating additional decoy items and strategically place them on a result page. Previously conducted user studies have shown that the level of confidence of recommender users can be increased by the addition of decoy items on recommender result pages.

The following paper is structured as follows: *Section 2* introduces a model which calculates dominance values and can be easily applied to utility systems like MAUT.

Section 3 presents a tool which implements the earlier described model and thus calculates dominance values. *Section 4* compares calculated dominance values with the choice distributions of a previously conducted decoy study. In Section 5 some *Conclusions* are drawn.

2 Calculating Choice-Set-Dependent Dominance

Calculating item utilities independently from the presented item set has already been widely used in utility-based recommender systems. Multi-Attribute-Utility-Theory (MAUT) is a very common approach to accomplish this task. Fig. 3 shows the principle for a MAUT base with two interest dimensions (*dim1, dim2*), and only two item- and customer properties. The *overall utility* is a sum of interest dimension specific *sub-utilities*. Those sub-utilities are products of the summations of customer property scores (*c1, c2*) and item property scores (*i1, i2*). The better and the more important the value of a certain property is for a certain dimension, the higher is the corresponding score (typically 1-10). If there is no information about the customer/user the item specific scores (i1, i2) can be multiplied with static weights instead of c1 and c2.

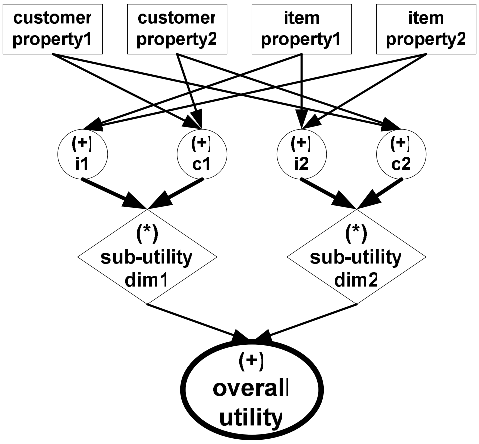


Fig. 3. Multi-Attribute-Utility-Theory (MAUT)

The *Simple Dominance Model* (SDM), which was first presented in [6], is able to calculate choice set dependent dominance values for items that are described by a certain number of numerical item properties. The very simple structure of the model demands linear property value functions. This means that the model cannot handle thresholds (e.g. a digital camera with a resolution lower than 2 mpix has no utility) or even more complex functions (e.g. step functions). In contrast MAUT uses property values which are mapped to linear scorings and is thus able to handle more complex value functions. A combination of both results in a model which calculates choice-set dependent dominance values for items using a flexible MAUT base as grounding. (1) is listing the basic formula of the SDM on top of MAUT.

$$DV_{x \in Set} = \frac{\sum_{s \in \{Set-x\}} \sum_{d \in Dimensions} weight_d * \sqrt{\left|\frac{x_d - s_d}{d_{\max} - d_{\min}}\right|} * \frac{x_d - s_d}{|x_d - s_d|}}{\#(Set-x)} \quad (1)$$

The dominance value *DV* for an item *x* out of a choice set *Set* compared to all other items in the Set (*Set-x*) is the sum of weighted (*weight$_d$*) differences of item scorings (x_d - s_d) for every sub-dimension. The differences of item scorings (x_d - s_d) are set in relation to the extreme scoring values of the set in that dimensions (d_{max} – d_{min}). The square root implicates that the similarity of items influences the dominance values. $(x_d–s_d)/|x_d–s_d|$ evaluates to 1 or -1 and only serves for preserving the correct sign. Compared to the MAUT scheme in Fig. 3 x_d and s_d correspond to instances of *i1* and *i2*, *d* corresponds to *dim1* and *dim2*, and *weight* corresponds to *c1* and *c2*. Simple example (see Fig. 4): Given is a simple choice set of three items (*target (T), competitor (C), decoy (D)*) . Each item has a certain number of properties which result in item scorings of two dimensions (*quality, price*).

- target (quality=6, price=2)
- competitor (quality=2, price=6)
- decoy (quality=5, price=1)

Each dimension should have a static weight of 0.5 (out of simplicity reasons).

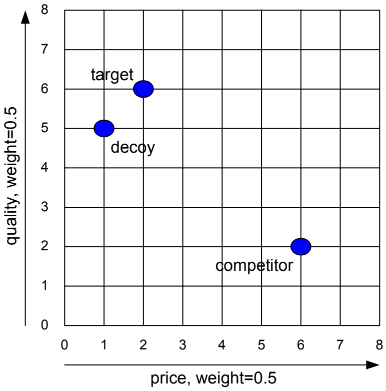


Fig. 4. Example for SDM

Applying the formula in (1), the dominance values for the items are calculated as follows:

DV_T=

$$\frac{0.5*\left\langle\sqrt{\frac{6(T;quality)-2(C;quality)}{6(T;quality)-2(C;quality)}}+\sqrt{\frac{6(T;quality)-5(D;quality)}{6(T;quality)-2(C;quality)}}\right\rangle+0.5*\left\langle\sqrt{\frac{2(T;price)-1(D;price)}{6(C;price)-1(D;price)}}-\sqrt{\frac{6(C;price)-2(T;price)}{6(C;price)-1(D;price)}}\right\rangle}{2}$$

=0.263

DV_C=

$$\frac{0.5*\left\langle-\sqrt{\frac{6(C;quality)-2(T;quality)}{6(T;quality)-2(C;quality)}}-\sqrt{\frac{5(C;quality)-2(D;quality)}{6(T;quality)-2(C;quality)}}\right\rangle+0.5*\left\langle\sqrt{\frac{6(C;price)-2(T;price)}{6(C;price)-1(D;price)}}+\sqrt{\frac{6(C;price)-1(D;price)}{6(C;price)-1(D;price)}}\right\rangle}{2}$$

=0.007

DV_D=

$$\frac{0.5*\left\langle-\sqrt{\frac{6(T;quality)-5(D;quality)}{6(T;quality)-2(C;quality)}}+\sqrt{\frac{5(D;quality)-2(C;quality)}{6(T;quality)-2(C;quality)}}\right\rangle+0.5*\left\langle-\sqrt{\frac{2(T;price)-1(D;price)}{6(C;price)-1(D;price)}}-\sqrt{\frac{6(C;price)-1(D;price)}{6(C;price)-1(D;price)}}\right\rangle}{2}$$

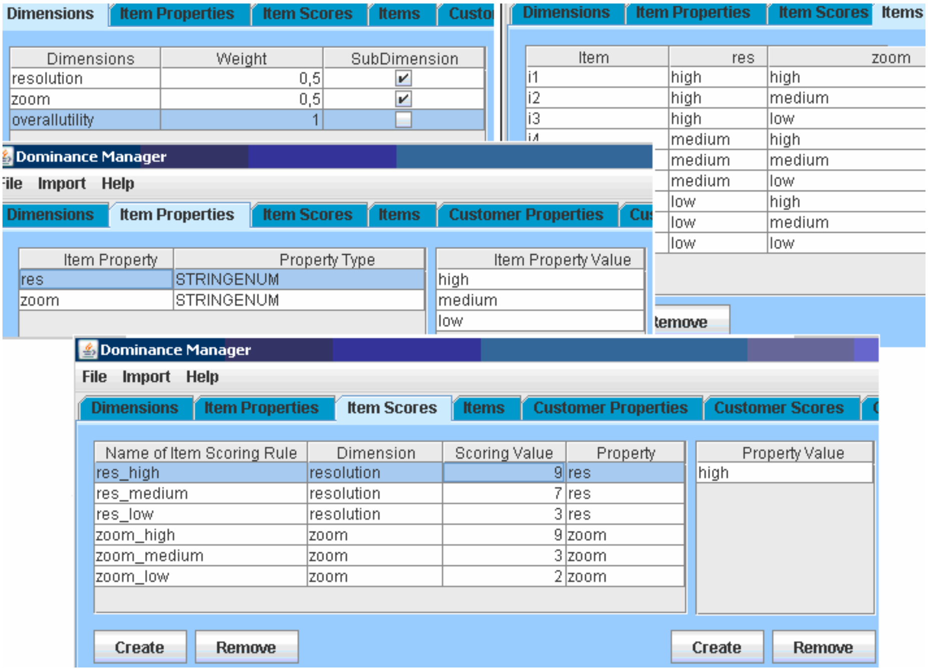
=-0.27

When the weights sum up to 1 the dominance values are between 1 and -1. The sum of the dominance values is always 0. Please note that in the example above a set consisting only of the target and the competitor would produce dominance values equal to 0 for both items. The addition of the decoy, which constitutes an asymmetrical dominated alternative, increases the dominance value of the target.

3 The Dominance Manager

The Dominance Manager is a tool which allows maintaining MAUT bases including interest dimensions, item/customer properties, and corresponding scores. Furthermore, it fully implements the dominance model discussed in the last section and thus allows calculating and visualizing choice-set dependent dominance values. Additionally to the direct input by hand, complete MAUT bases can be imported from Recommenders [5]. Fig. 5 shows the *Dimensions* tab: Interest dimensions can have sub dimensions. Typically there is only one root dimension (*overallutility*), which holds the overall utility (compare Fig. 3) and all other dimensions are sub dimensions of that root dimension. The static dimension weights can be used for utility and dominance calculation if no valuable customer information (i.e. customer scores) is known. Fig. 7

also shows the screenshots of the *Item Properties-, Items Properties-, and Items tab*. If the type is an enum-type, the possible values have to be entered and additionally a score for every value has to be defined. Alternatively the Dominance Manager supports complex numerical scoring functions. To this end the edge points of the function have to be defined. For example a linear floating point function requires two edge points and their corresponding scorings. Scorings for values between the edge points are calculated by linear interpolation. In the *Items* tab items can be created, removed and altered. Every item is described by all item properties. Customers are handled by the Dominance Manager similar to items. Therefore also tabs for *Customer Properties (and values), Scorings, and Customer* instances are found in the Dominance Manager (Fig. 6).



Dimensions	Weight	SubDimension
resolution	0,5	☑
zoom	0,5	☑
overallutility	1	□

Item	res	zoom
i1	high	high
i2	high	medium
i3	high	low
i4	medium	high
	medium	medium
	medium	low
	low	high
	low	medium
	low	low

Item Property	Property Type
res	STRINGENUM
zoom	STRINGENUM

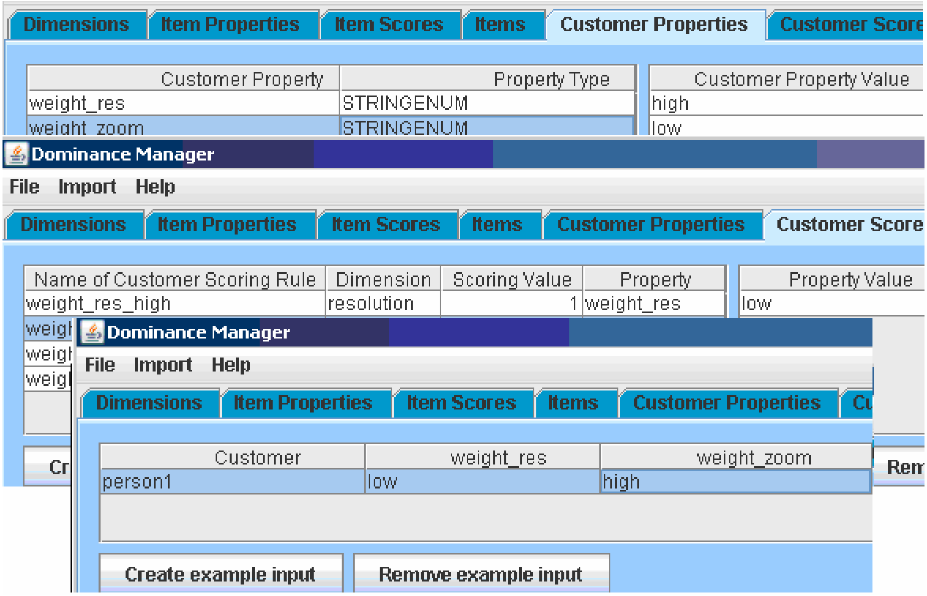
Item Property Value
high
medium
low

Name of Item Scoring Rule	Dimension	Scoring Value	Property
res_high	resolution	9	res
res_medium	resolution	7	res
res_low	resolution	3	res
zoom_high	zoom	9	zoom
zoom_medium	zoom	3	zoom
zoom_low	zoom	2	zoom

Property Value
high

Fig. 5. Items, Properties, Scorings tabs and the Dimension tab

Combining all information about dimensions, items, and customers the Dominance Manager calculates summations of the scorings of items and customers and resulting (sub-) utilities of items which can be for a specific customer or basing on static weights of interest dimensions (Fig.7). Implementing the dominance model discussed in the last section, the Dominance Manager allows defining sets of items and calculates resulting choice-set dependent dominance values. The dominance values can additionally be presented in chart form (Fig. 7).



Customer Property	Property Type
weight_res	STRINGENUM
weight_zoom	STRINGENUM

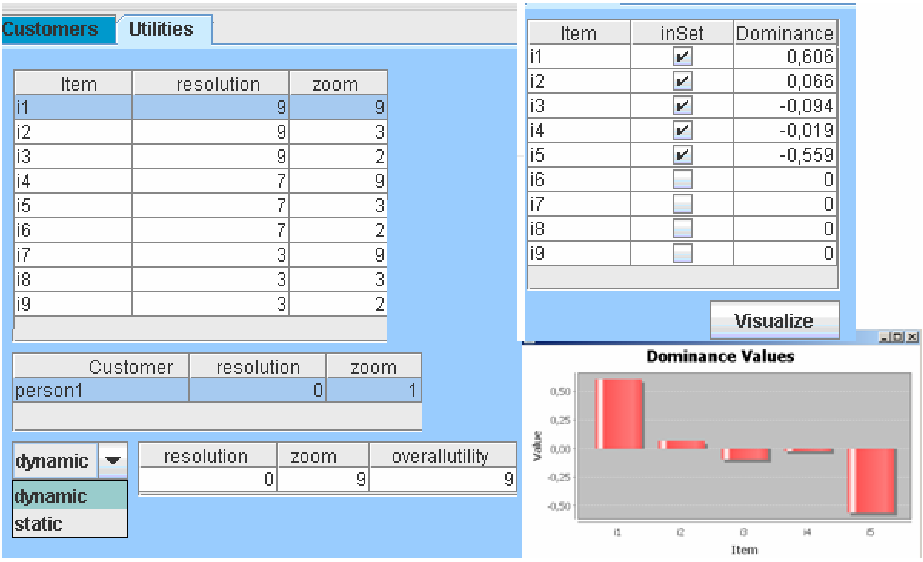
Customer Property Value
high
low

Name of Customer Scoring Rule	Dimension	Scoring Value	Property
weight_res_high	resolution	1	weight_res
weigh			
weigh			
weigl			

Property Value
low

Customer	weight_res	weight_zoom
person1	low	high

Fig. 6. Customers, Properties, Scorings tabs



Item	resolution	zoom
i1	9	9
i2	9	3
i3	9	2
i4	7	9
i5	7	3
i6	7	2
i7	3	9
i8	3	3
i9	3	2

Customer	resolution	zoom
person1	0	1

resolution	zoom	overallutility
0	9	9

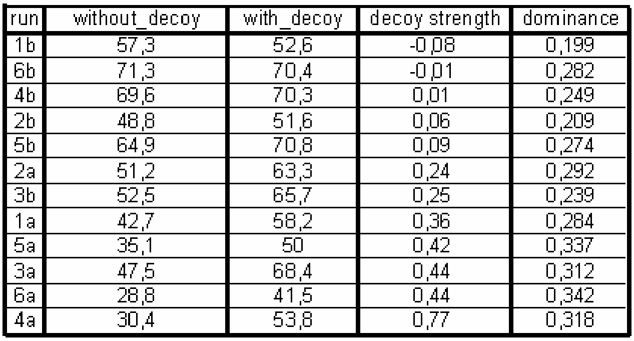
Item	inSet	Dominance
i1	☑	0,606
i2	☑	0,066
i3	☑	-0,094
i4	☑	-0,019
i5	☑	-0,559
i6	□	0
i7	□	0
i8	□	0
i9	□	0

Fig. 7. Calculation and visualization of utilities and dominance values

4 Empirical Evaluation

In order to investigate if the dominance values which are calculated by the Dominance Manager are in line with empirical findings, we fed the Dominance Manager with item information which was used for a previously conducted decoy study (details can be found in [10]). One of the goals of the study was to replicate the asymmetrical dominance effect in the domain of hotel rooms. The rooms were described in terms of *price* and *space* and participants were asked to choose the option with the *best price-performance ratio*. Altogether there were 535 participants, which led to 1585 relevant choices (the study also investigated aspects which are not relevant for this paper). The choice sets were consisting either only of a core set (room A and room B, i.e. control group) or additionally of a decoy room which was asymmetrically dominated by room A or room B. Furthermore, the choice sets were different in terms of price segment, i.e. there were 6 different segments ranging from cheap to expensive hotel rooms. To be able to compare dominance values calculated by the Dominance Manager with the empirical findings we had to calculate the relative decoy strength in the hotel room study. We define *the relative decoy strength* as the percentage of target bookings in presence of a decoy minus the percentage of target bookings without a decoy divided by the percentage of target bookings without a decoy, i.e. *(with_decoy - without_decoy)/without_decoy* (see Table 1). Table 1 summarizes the calculated decoy strengths and the corresponding dominance values calculated by the Dominance Manager[1]. Altogether there were 12 decoy groups (6 segments, decoy for room A/roomB) which were confronted to the corresponding control group (i.e. *without_decoy*). It is obvious that the lowest decoy strengths in the user study correspond to the lowest dominance values whereas conditions where the decoy was performing well also produced highest dominance values. A corresponding correlation calculation was highly significant (Pearson Chi² = .75, p < .001). Additionally to the user interface which serves mainly as support of knowledge engineers and salesmen the Dominance Manager also provides an interface such that recommender systems can use dominance information automatically.

Table 1. Relative decoy strength compared with the calculated dominance values



run	without_decoy	with_decoy	decoy strength	dominance
1b	57,3	52,6	-0,08	0,199
6b	71,3	70,4	-0,01	0,282
4b	69,6	70,3	0,01	0,249
2b	48,8	51,6	0,06	0,209
5b	64,9	70,8	0,09	0,274
2a	51,2	63,3	0,24	0,292
3b	52,5	65,7	0,25	0,239
1a	42,7	58,2	0,36	0,284
5a	35,1	50	0,42	0,337
3a	47,5	68,4	0,44	0,312
6a	28,8	41,5	0,44	0,342
4a	30,4	53,8	0,77	0,318

[1] Corresponding weights for the dimensions *price* and *space* were set to 0.5.

5 Conclusions

The Dominance Manager presented in this paper is the first tool which allows calculating and visualizing decoy effects in the context of MAUT based recommender systems. As a basis for the calculation serves the *Simple Dominance Model* which was adapted in this paper in order to be combined with MAUT utility bases. The Dominance Manager is therefore a utility-based tool which allows the detection of unforeseen decoy effects, detection of situations where a decoy could alleviate decision taking, and identification of strong decoy items which serve the aim of alleviating decision taking on result pages of recommender systems.

References

1. Ariely, D., Wallsten, T.: Seeking subjective dominance in multidimensional space: An exploration of the asymmetric dominance effect. Organizational Behaviour and Human Decision Processes 63(3), 223–232 (1995)
2. Burke, R.: Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction 12(4), 331–370 (2002)
3. Dhar, R., Glazer, R.: Similarity in context: Cognitive representations and violations of preference and perceptual invariance in consumer choice. Organizational Behaviour and Human Decision Processes 67, 280–293 (1996)
4. Felfernig, A., Gordea, S., Jannach, D., Teppan, E., Zanker, M.: A Short Survey of Recommendation Technologies in Travel and Tourism. ÖGAI Journal 4(25), 17–22 (2006)
5. Felfernig, A., Gula, B., Teppan, E.: Knowledge-based recommender technologies for marketing and sales. International Journal of Pattern Recognition and Artificial Intelligence 21, 333–355 (2007)
6. Felfernig, A., Gula, B., Leitner, G., Maier, M., Melcher, R., Schippel, S., Teppan, E.: A Dominance Model for the Calculation of Decoy Products in Recommendation Environments. In: Proceedings of the AISB Convention, vol. (3), pp. 43–50 (2008) ISBN: 902956621
7. Kahneman, D.: Maps of bounded rationality: psychology for behavioral economics. The American Economic Review 93(5), 1449–1475 (2003)
8. Simonson, I.: Choice Based on Reasons: The Case of Attraction and Compromise Effects. Journal of Consumer Research 16 (1989)
9. Simonson, I., Tversky, A.: Choice in context: Tradeoff contrast and extremeness aversion. Journal of Marketing Research 39, 281–292 (1992)
10. Teppan, E., Felfernig, A.: Der Asymmetrische Dominanzeffekt und seine Bedeutung für E-Tourismus Plattformen. In: Proceedings of Wirtschaftsinformatik Conference 2009, pp. 791–800 (2009)
11. Winterfeldt, D., Edwards, W.: Decision Analysis and Behavioral Research. Cambridge University Press, Cambridge (1986)

An Evolutionary Algorithm with Non-random Initial Population for Path Planning of Manipulators

Chien-Chou Lin

Department of Computer Science and Information Engineering,
National Yunlin University of Science and Technology
No. 123, University Road, Section 3, Douliou, Yunlin 64002, Taiwan, R.O.C.
linchien@yuntech.edu.tw

Abstract. In this paper, a hierarchical evolutionary algorithm is proposed for the path planning of manipulators. The proposed algorithm consists of a global path planner (GPP) and a local motion planner (LMP). The global planner, a MAKLINK based approach, plans a trajectory for a robot end-effector from a starting free-space to goal free-space. An evolutionary algorithm with a non-random initial population is adopted to plan the manipulator configurations along a path given by the former stage. Once the optimal configuration is obtained by the evolutionary algorithm, the optimal chromosomes will be reserved as the initial population. Since the initial population is non-random, the evolution is more efficient and the planned path is smoother than traditional GA. Simulation results show that the proposed algorithm works well, specifically in terms of collision avoidance and computation efficiency.

Keywords: Evolutionary algorithm, path planning, motion planning, collision-avoidance.

1 Introduction

Path planning of a manipulator is focused on determining a collision-free trajectory from its original location and orientation (called starting configuration) to goal configuration [1], as shown in Fig. 1. In recent years, many robot path planning algorithms have been proposed [1-28]. In general, this research adopted two basic approaches, configuration space (c-space) based approach [3-8] and geometric based algorithms [9-23]. The c-space based approach considers both the manipulator and the obstacles at the same time by identifying manipulator configurations intersecting the obstacles. A point in a c-space indicates that configuration of a manipulator. A configuration is usually encoded by a set of manipulator's parameters; i.e. angles of links of manipulators. The forbidden regions in the c-space are the points which imply manipulator configurations intersecting the obstacles. Thus, path planning is reduced to the problem of planning a path from a start point to a goal in free space [3].

Unlike the configuration space based approaches, the geometric algorithms directly use spatial occupancy information of the workspace to solve path planning problems. Workspace-based algorithms usually extract relevant information about the free space and use them together with the robot geometry to find a path. In addition to

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 193–201, 2009.

collision avoidance, some approaches try to find paths with minimum risk of collision. To minimize such a risk, repulsive potential fields between robots and obstacles are used in [16], [17], [18], [19], [20] and [21] to match their shapes in the path planning.

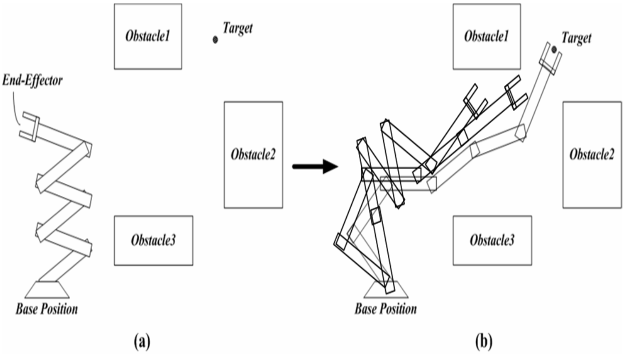


Fig. 1. Path planning of a five-link manipulator. (a) the initial configuration. (b) a 3-configuration path to goal.

Recently, some GA-based algorithms [24-28] have been proposed for robot path planning since the searching of GA algorithms is complete. However, the GA-based approaches are slower because they are tools of evolution and not as applicable for specific optimizations. In this paper, we propose a hierarchical evolutionary algorithm for path planning of manipulators to improve the performance of a GA-based planner. The proposed algorithm includes a global path planner (GPP) and a local motion planner (LMP). The global planner, adopting the *MAKLINK* approach, plans a trajectory for robot end-effector from a starting free-space to a goal free-space, as discussed in Section 2. The evolutionary algorithm with a non-random initial population is adopted to plan the manipulator configurations along the path given by the former stage, as shown in Section 3. In Section 4, simulation results are presented for path planning of manipulators in different 2-D workspaces. Section 5 gives a summary of this work.

2 Global Path Planning

Because of coupling global planning and local planning, many existing algorithms require complex searching and suffer from local minimum problems. The proposed hierarchical algorithm consists of global planner and local planner. The former determines the primary moving direction of manipulators and the latter is a motion planner deriving a manipulator configuration with a given position of its end-effector. Details of the two planning algorithms will be introduced separately in next section.

Global path planning can be simplified as a planner for point robots. In this paper, the *MAKLINK* [2] based on free link concepts is adopted since it is easy to implement and has been shown to work well in many cases. In [2], the key idea of *MAKLINK* is the division of the free workspace into several free spaces with free links. The free space is structured as a free convex area using the developed free link approach, as shown in Fig. 2.

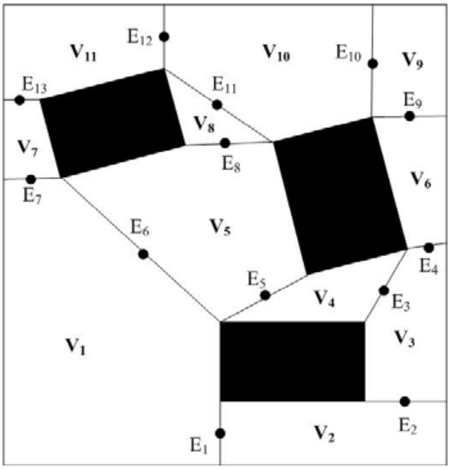


Fig. 2. Free links division of free workspace into several free spaces

First, we connect all vertices of an obstacle to other obstacles and workspace boundaries, except if those connections intersect with any edge of the obstacles. Free links of this obstacle are selected from these connections according to the following steps. Second, the lines of a corner of an obstacle are sorted according to their length, from the shortest to the longest. Third, checking the outside angles of the current corner results from the current line connection, the line is decided as the best free link if the two angles are less than or equal to 180 degrees. Other links to this corner are ignored. Notably, if one of the two angles is more than 180 degrees, the link is a free link and will be added to the list of free links. For such a corner, other links are needed to divide an outside angle which is more than 180 degrees. Therefore, the shortest links are selected sequentially from the free link list to divide the outside angles to less than 180 degrees.

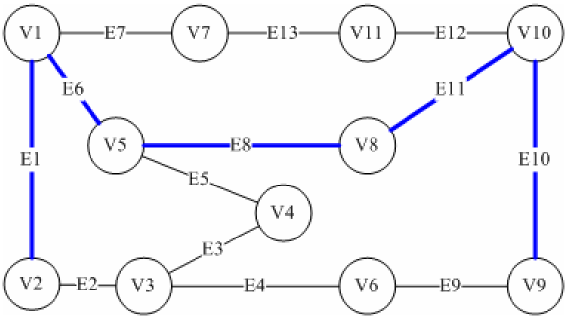


Fig. 3. The graph of Fig. 2

The selection of the best free links and free links is repeated for all obstacles. As shown in Fig. 2, the free workspace is divided into several free spaces, V_1, V_2... and V_{11}. The segmented workspace of Fig. 2 can be represented as a graph, $G=\{V_i,E_k\}$, shown in Fig. 3. Free spaces are represented as vertices (V_i) of G and free links are represented as edges (E_i) of G. The global path from start (V_2) to goal (V_{10}) is obtained using a simple search algorithm. The global path (E_1, E_6, E_8, E_{11}) is shown as a blue path in Fig. 3 and as a dash-line path in Fig. 4.

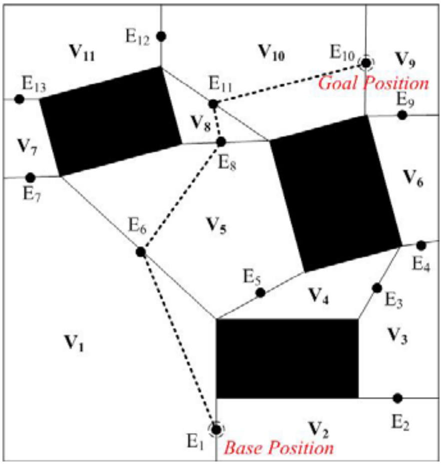


Fig. 4. The planned global path of Fig. 3

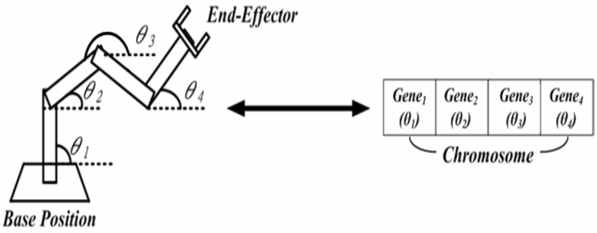


Fig. 5. The configuration of a manipulator can be represented as a chromosome and the genes are its joint angles

3 Local Planning

The global planner determines a path (E_1, E_6, E_8, E_{11}) that the end-effector should follow to the goal. In order to obtain fine trajectories, more points are interpolated within a free space. Thus, a finer path is given as (q_1, q_2, q_3,…,q_n). For each position, the evolutionary algorithm based local planner should plan a configuration for the manipulator to which the end-effector is fixed at the point.

Basically, the evolutionary algorithm is used for searching optimal solutions through the whole solution space and thus can avoid being trapped in local minima. In order to use an evolutionary algorithm optimizing the configuration of a manipulator, the joint angles of the manipulator configuration are encoded into one chromosome in this paper. A configuration of a k-link manipulator is denoted as its joint angles$(\theta_1, \theta_2,\ldots, \theta_k)$. Each joint angle is represented as a gene of a chromosome. For each chromosome, the Cartesian coordinates of the end-effector can be determined. The chromosome is then constructed as shown in Fig. 5. The proposed evolutionary algorithm can be formed as below:

Algorithm Non-Random-Initial-GA

```
Begin
  i = 1;  /* Initial the first intermediate goal */
  t = 0;  /* Initial the evolutionary generations */
  Randomly Generate an initial population $P_i$ ( t );
  fitness($P_i$(t));
  repeat until (reach the final goal $q_n$) Do
            $P_{i+1}$(t)= $P_i$(t);
      repeat until (reach the intermediate goal $q_i$) Do
        select $P_i$(t+1) from $P_i$(t);
        crossover($P_i$(t+1));
        mutate ($P_i$ (t+1) );
        fitness($P_i$(t+1));
        t = t + 1;
      end
      i= i +1;
            end
End
```

3.1 Initializing Population

While the population, $P_1(0)$, of the first intermediate goal is generated randomly, the initial populations ($P_i(0)$, $i>1$) of other intermediate goals are derived from the last generation of the preceding intermediate goal. Since these initial populations are eugenic and are inheriting from the ancestor, the evolution time is reduced and the obtained path of the manipulator is smoother.

3.2 Evaluation Criteria

Generally, selection is performed according to the fitness of every chromosome and where fitness evaluation of EA is an objective function for chromosomes. In our design, an optimal chromosome, a configuration, should be collision-free and should reach the goal. Therefore, the fitness function can be defined as:

$$V_f = \begin{cases} D_s, & \textit{if no collision} \\ V_{\max}, & \textit{if collision} \end{cases} \tag{1}$$

where V_f is cost function. When a configuration collides with obstacles, the V_f is equal to V_{max} which is a punish cost. The cost of no-collision configuration (D_s) is defined as (2), the distance between the end-effector and intermediate goals.

$$D_s = \sqrt{(G_x - P_x)^2 + (G_y - P_y)^2} \tag{2}$$

where *(G$_x$, G$_y$)* is the coordinator of the intermediate goal and *(P$_x$, P$_y$)* is the coordinator of the end-effector defined as

$$P_x = \sum_{i=0}^{dof} x_i + \cos\theta_i \times L_i$$
$$P_y = \sum_{i=0}^{dof} y_i + \sin\theta_i \times L_i \tag{3}$$

where L_i is the length of link *i*.

3.3 Crossover and Mutation Operators

The crossover of the traditional GA is exchanging some genes of a pair of chromosomes randomly. In this paper, since the *i*-th gene of a chromosome represents the *i*-th link of a manipulator, the crossover operator is defined by the notion of exchanging similarly positioned genes of a pair of chromosomes.

For manipulator motion, a small change in the angle of the base link will cause much greater manipulator movements further away from the base. Therefore, the probability of mutation for every links is different. The distal link has the biggest probability of mutation and the base link has the smallest one.

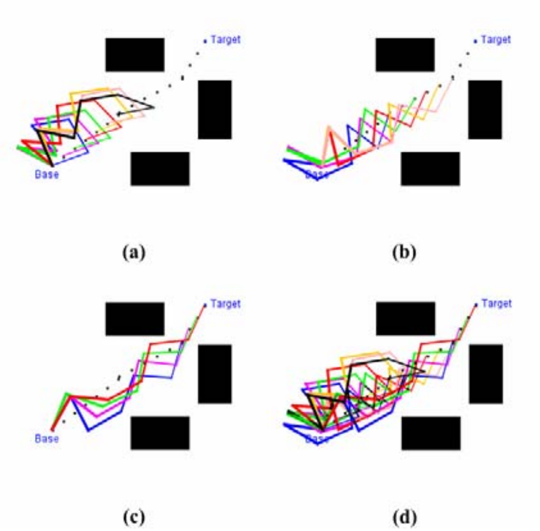


Fig. 6. A path planning example of a 6-link manipulator

4 Experimental Results

In this section, simulation results are presented for path planning implemented by JGAP API. Figs. 6(a-c) show partial trajectories of a 6-link manipulator. Fig. 6(d) shows the whole planned trajectory. The population is 100 and the maximum generation is 600. There is 10% of the initial population that is non-random in our simulations.

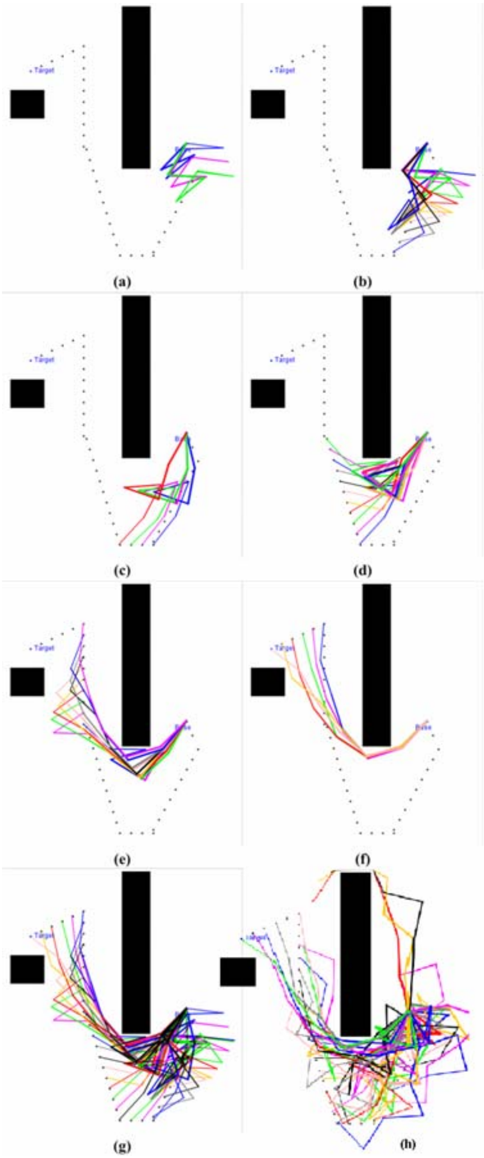


Fig. 7. (a)-(f) A sequence of partial paths of another 6-link manipulator example. (g) The complete path obtained by proposed algorithm. (h) The complete path obtained by traditional GA.

The partial and whole trajectories of another example of a 6-link manipulator are shown in Figs. 7(a)-(f) and Fig. 7(g), respectively. It is easy to see that the obtained trajectory is smooth. The whole trajectory obtained by traditional GA is shown in Fig.7(h). Since the all populations are randomly generated, some unsuitable configurations are derived, e.g., the three configurations over the big obstacle.

5 Conclusions

A hierarchical EA path planner for hyper-redundant manipulators is proposed in this paper. Unlike most planning algorithms for manipulators considering path planning and motion planning together, the proposed algorithm consists of a global path planner (GPP) and a local motion planner (LMP). Put simply, the GPP plans a trajectory for a robot's end-effector and the LMP plans manipulator configurations along the path. Since the proposed algorithm processes path planning and motion planning individually, the complexity of the proposed EA is reduced. Furthermore, since the initial population of EA is non-random, the continuity of the manipulator's configurations is reserved and the obtained trajectories are smoother. Simulation results show that the proposed algorithm is able to plan a smooth trajectory to the goal while reducing evolution time. In addition, the unfeasible configurations which may be obtained by traditional GA can be avoided as shown in simulations of this study.

Acknowledgments. This work is supported by National Science Council of Taiwan under Grant No. NSC 95-2221-E-366-010 and NSC 97-2221-E-366-004.

References

1. Reif, J.H.: Complexity of the mover's problem and generalizations. In: Proceedings of Foundations of Computer Science, pp. 421–427 (1979)
2. Habib, M.K., Asama, H.: Efficient method to generate collision Free paths for an autonomous mobile robot based on new Free space structuring approach. In: Proceedings of IEEE/RSJ International Workshop on Intelligent Robots and Systems, vol. 2, pp. 563–567 (1991)
3. Lozano-Perez, T., Wesley, M.A.: An Algorithm for planning collision-Free paths among polyhedral obstacles. Communications of the ACM 22, 560–570 (1979)
4. Lozano-Perez, T.: Spatial Planning: a configuration space approach. IEEE Transactions on Computers C-32, 108–120 (1983)
5. Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.H.: Probabilistic Roadmap for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics & Automation 12(4), 566–580 (1996)
6. Lozano-Perez, T.: A simple motion-planning algorithm for general robot manipulator. IEEE Journal of Robotics and Automation 3(3), 224–238 (1987)
7. LaValle, S.M.: Rapidly-exploring random tree: A new tool for path planning. Technology Report, no. 98–11 (1998)
8. Kuffner, J.J., LaValle, S.M.: RRT-connect: An efficient approach to single-query path planning. In: Proceedings of. IEEE International Conference on Robotics and Automation, pp. 995–1001 (2000)
9. Laliberte, T., Gosselin, C.: Efficient algorithms for the trajectory planning for redundant manipulators with obstacle avoidance. In: Proceedings of IEEE International Conference on Robotics and Automation, vol. 3, pp. 2044–2049 (1994)
10. Ralli, E., Hirzinger, G.: Fast path planning for robot manipulators using numerical potential fields in the configuration space. In: Proceedings of the IEEE/RSJ/GI Intl. Conf. on Intelligent Robots and Systems, Advanced Robotic System and the Real World, vol. 3, pp. 1922–1929 (1994)

11. Barraquand, J., Lotombe, J.C.: Robot motion planning: a distributed representation approach. International Journal of Robotics Research 10(6), 628–649 (1991)
12. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. International Journal of Robotics Research 5(1), 90–98 (1986)
13. Latombe, J.C.: Motion planning: A journey of robots, molecules, digital actors, and other artifacts. International Journal of Robotics Research 18(11), 1119–1128 (1999)
14. Barraquand, J., Langois, B., Latombe, J.C.: Numerical potential field techniques for robot path planning. IEEE Transactions on Robotics and Automation 22, 224–241 (1992)
15. Masoud, A.A., Bayoumi, M.M.: Intercepting a maneuvering target in a multidimensional stationary environment using a ware equation potential field strategy. In: IEEE International Symposium on Intelligent Control, pp. 243–248 (1994)
16. Chuang, J.-H., Lin, C.-C., Hsieh, C.-T.: A Potential-Based Path Planning of Articulated Robots with 2-DOF joints. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1827–1832 (2005)
17. Lin, C.-C., Pan, C.-C., Chuang, J.-H.: A Novel Potential-Based Path Planning of 3-D Articulated Robots with Moving Bases. Robotica 22(4), 359–367 (2004)
18. Chuang, J.-H., Lin, C.-C., Kuo, L.-W.: Potential-Based Path Planning for Robot Manipulators. In: Proceeding of IEEE International Conference on Method and Models in Automation and Robotics, pp. 1031–1036 (2002)
19. Lin, C.-C., Chuang, J.-H.: Potential-Based Path Planning for Robot Manipulators in 3D Workspace. In: Proceedings of. IEEE International Conference on Robotic and Automation, pp. 3353–3358 (2003)
20. Lin, C.-C., Pan, C.-C., Chuang, J.-H.: A Novel Potential-Based Path Planning of 3-D Articulated Robots with Moving Bases. In: Proceedings of IEEE International Conference on Robotic and Automation, pp. 3365–3370 (2004)
21. Chuang, J.-H.: Potential-Based Modeling of Three-Dimensional Workspace for Obstacle Avoidance. IEEE Transactions on Robotics and Automation 14(5), 778–785 (1998)
22. Brooks, R., Lozano-Perez, T.: A subdivision algorithm in configuration space for findpath with rotation. IEEE Transcations on System, Man & Cybernetics 15(2), 224–233 (1985)
23. Aron, O., Lozano-Perez, T.: Visible Decomposition: Real-Time path Planning in Large Planar Environments. AI Memo 1638 (1996)
24. Solterio-Pires, E.J., Tenrerio-Machado, J.A.: A trajectory planner for manipulators using genetic algorithm. In: Proceedings of the IEEE International Symposium on Assembly and Task Planning, pp. 163–168 (1999)
25. Parker, J.K., Khoogar, A.R., Goldberg, D.E.: Inverse Kinematics of redundant robots using genetic algorithms. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 271–276 (1989)
26. Khoogar, A.R., Parker, J.K.: Obstacle avoidance of redundant manipulators using genetic algorithms. In: Proceedings of IEEE International Conference on Robotic and Automation, pp. 317–320 (1991)
27. Chen, H., Du, X., Gu, W.-K.: Global Path Planning Based on Neural Network and Genetic Algorithm in a Static Environment. Artificial Neural Networks, 34–42 (2004)
28. Gill, M., Zomaya, A.: A parallel collision-avoidance algorithm for robot manipulators. IEEE Concurrency 6(1), 68–78 (1998)

An Improved Particle Swarm Optimization with Feasibility-Based Rules for Constrained Optimization Problems*

Chao-li Sun[1], Jian-chao Zeng[1], and Jeng-shyang Pan[2]

[1] Complex System and Computational Intelligence Laboratory,
Taiyuan University of Science and Technology, Taiyuan, Shanxi, P.R. China, 030024
clsun1225@163.com
[2] Department of Electronic Engineering,
National Kaohsiung University of Applied Sciences, Kaohsiung, 807, Taiwan
jspan@cc.kuas.edu.tw

Abstract. This paper presents an improved particle swarm optimization (IPSO) to solve constrained optimization problems, which handles constraints based on certain feasibility-based rules. A turbulence operator is incorporated into IPSO algorithm to overcome the premature convergence. At the same time, a set called FPS is proposed to save those P_{best} locating in the feasible region. Different from the standard PSO, g_{best} in IPSO is chosen from the FPS instead of the swarm. Furthermore, the mutation operation is applied to the P_{best} with the maximal constraint violation value in the swarm, which can guide particles to close the feasible region quickly. The performance of IPSO algorithm is tested on a well-known benchmark suite and the experimental results show that the proposed approach is highly competitive, effective and efficient.

Keywords: Particle swarm optimization, Feasibility-based rules, constrained optimization problems.

1 Introduction

Evolutionary algorithms such as Genetic Algorithm, Evolutionary Strategies, Evolutionary Programming, etc. have been proposed to handle optimization problems [1-5]. Besides, many of them have been successfully applied for tackling constrained optimization problems during the past few years. Particle Swarm Optimization (PSO) is a new global evolutionary algorithm proposed by Kennedy and Eberhart in 1995 [6,7], its idea was based on the simulation of simplified social models such as bird flocking and fish schooling. PSO has been successfully applied in a variety of fields mainly for unconstrained continuous optimization problems [22-25]. Yet many real-world applications involve difficult constrained optimization problems that must be solved

* This work was supported in part by National Science Foundation of China under Grant NO.60674104 and by Natural Science Foundation of Shanxi Province of China under Grant NO. 20081030.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 202–211, 2009.

efficiently and effectively, such as engineering design, VLSI design, structural optimization, economics, locations and allocation problems [8]. Disliking those deterministic optimization approaches, such as Feasible Direction and Generalized Gradient Descent [8,9], PSO algorithm is generally independent of the mathematic characteristics of the objective problem, and has been considered a valid technique to solve constrained optimization problems with a simple concept, easy implementation, quick convergence. However, like other aforementioned stochastic evolutionary algorithms, PSO also need an explicit constraint-handling mechanism. Generally, three main constraint-handling mechanisms can be incorporated into PSO for solving constrained optimization problems.

The penalty function method has been the most popular constraint-handling technique due to its simple principle. It converts a constrained optimization problem to an unconstrained optimization one through adding a penalty item to the objective function [10]. This method may work quite well for some problems, but it requires a careful tuning of the penalty parameters, and which turns out to be a difficult optimization problem itself [12], since both under_ and over_ penalizations may result in an unsuccessful optimization.

The feasibility-based method introduces feasibility-based rules which give an instruction on the determination of the best solution of the population (g_{best}) and the best historical solution of every particle (P_{best}) into PSO [13-14,20]. There is no need to design additional parameters, but according to feasibility-based rules, feasible solutions are always considered better than infeasible ones, which may cause the over-pressure of selecting feasible solutions so as to result in premature convergence.

The constrained-preserving method (the feasible solutions method) reduces the search space by ensuring that all the candidate solutions satisfy the constraints at all times [11, 15-16]. Solutions are initialized within the feasible space, and transformations of candidate solutions are such that the resulting solutions still lie within the feasible region. This method requires an initialization of particle inside the feasible region, which may need a long time initialization process and may be hard to achieve for some problems.

This paper proposes an improved particle swarm optimization (IPSO) with feasibility-based rules to solve constrained optimization problems. A turbulence operator is introduced into IPSO algorithm to overcome the premature convergence. Otherwise, a set, called FPS, is specially introduced in IPSO algorithm to keep those P_{best} that is in the feasible region at current generation. Different to the standard PSO, g_{best} of IPSO is the best P_{best} from FPS, but the swarm. When the total number of P_{best} in FPS is less than the predefined constant, the mutation operation is manipulated to P_{best} that has the maximal constraint violation value in the swarm, which can guide particles to close the feasible region quickly.

The paper is organized as follows: In section 2, the problem of interest and the particle swarm optimization algorithm are stated briefly. Our proposed approach IPSO is provided in section 3. In section 4, the experimental setup and the results obtained are presented. Finally, section 6 presented the conclusions and the proposal for future research.

2 Basic Concepts

2.1 Problem Statement

Generally, a constrained problem can be described as follows:

$$\begin{array}{lll} min & f(\vec{x}) & \\ \text{s.t.} & g_i(\vec{x}) \le 0 & i = 1, 2, \cdots, m \\ & h_j(\vec{x}) \le 0 & j = 1, 2, \cdots, l \\ & x_{dmin} \le x_d \le x_{dmax} & d = 1, 2, \cdots, D \end{array} \quad (1)$$

Where $\vec{x} = (x_1, x_2, \cdots, D)$ is the vector of solutions such that $\vec{x} \in S \subseteq R^D$, S is defined as an D-dimensional space composed by lower and upper bounds $[x_{dmin}, x_{dmax}, d = 1, 2, \cdots, D$. m is the number of inequality constraints, l is the number of equality constraints, and the feasible region $F \subset S$ is the region of S that all constraints are satisfied. Commonly, an equality constraint is transformed into two inequality constraints $h_j(\vec{x}) \le \delta$ and $h_j(\vec{x}) \ge -\delta$, where δ is the tolerance allowed (a very small positive value). We call $\vec{x}$ a feasible solution when it satisfies all the constraints.

2.2 Standard Particle Swarm Optimization

Particle Swarm Optimization (PSO) is proposed as a global evolutionary algorithm by Kennedy and Eberhart in 1995[6,7], its idea was based on the simulation of simplified social models such as bird flocking and fish schooling. In PSO, it assumes that in a D-dimensional search space $S \subseteq R^D$, the swarm consists of N particles each has no volume and no weight, holds its own velocity and denotes a solution. The trajectory of each particle in the search space is dynamically adjusted by updating the velocity of each particle, according to its own flying experience as well as the experience of neighbor particles(built through tracking and memorizing the best position encountered). Particle i is in effect a D-dimensional vector $\vec{x}_i = (x_{i1}, x_{i2}, \cdots, x_{iD}) \in S$. Its velocity is also a D-dimensional vector $\vec{v}_i = (v_{i1}, v_{i2}, \cdots, v_{iD}) \in S$. The best historical position visited by particle i is a point in S, denoted as $P_i = (P_{i1}, P_{i2}, \cdots, P_{iD})$ or P_{best}, and the best historical position that the entire swarm has passed is denoted as $P_g = (P_{g1}, P_{g2}, \cdots, P_{gD})$ or g_{best}. To particle i, the new velocity and the new position of the d-th dimension ($1 \le d \le D$) are updated as follows[17]:

$$v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 (P_{id}(t) - x_{id}(t)) + c_2 r_2 (P_{gd}(t) - x_{id}(t))) \quad (2)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (3)$$

Where ω is a parameter called the inertia weight, c_1 and c_2 are positive constants respectively referred as cognitive and social parameters, r_1 and r_2 are random numbers uniformly distributed in [0,1].

The process is repeated until a user-defined stopping criterion is reached. For more detail the reader is referred to [18].

2.3 The Feasibility-Based Rule

Referring to [13], feasibility-based rules employed in this paper are described as follows:

(1) Any feasible solution is preferred to any infeasible solution.
(2) Between two feasible solutions, the one having better objective function value is preferred.
(3) Between two infeasible solutions, the one having smaller constraint violation is preferred.

Based on the above criteria, in the first and the third cases the search tends to the feasible region rather than infeasible one, and in the second case the search tends to the feasible region with good solutions. In brief, such a simple rule aims at obtaining good feasible solutions.

3 Our Proposed Approach

Figure 1 shows the algorithm of our IPSO algorithm.

Comparing to the standard particle swarm optimization, our algorithm is different from the standard PSO in following two aspects.

3.1 Updating P_{best} and g_{best}

In this paper, the constraint violation value of an infeasible solution is calculated as follows:

$$viol(x) = \sum_{i=0}^{m} max(0, g_i(\vec{x})) + \sum_{j=0}^{l} max(0, abs(h_j(\vec{x}))) \tag{4}$$

Suppose that $P_i(t)$ represents P_{best} of particle i at generation t and $\vec{x}_i(t+1)$ represents the newly generated position of particle i at generation $t+1$. In the standard PSO, $\vec{P}_i(t+1) = \vec{x}_i(t+1)$ only if $f(\vec{P}_i(t)) > f(\vec{x}_i(t+1))$. While in IPSO algorithm, the feasibility-based rule is employed. That is, $P_i(t)$ will replaced by $\vec{x}_i(t+1)$ at any of the following scenarios:

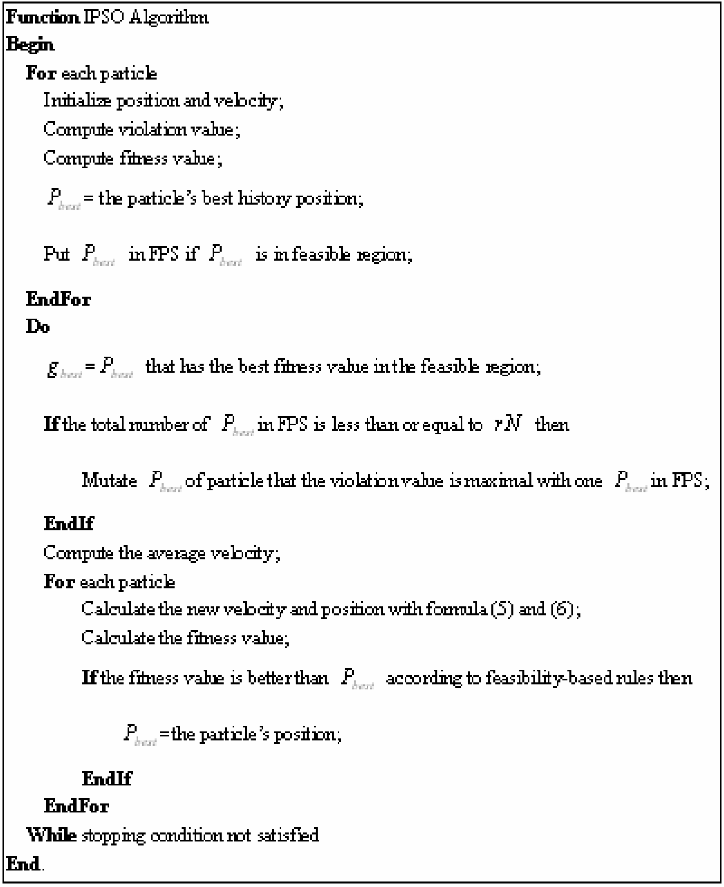
(1) $\vec{P}_i(t)$ is infeasible, but $\vec{x}_i(t+1)$ is feasible.
(2) Both $\vec{P}_i(t)$ and $\vec{x}_i(t+1)$ are feasible, but $f(\vec{P}_i(t)) > f(\vec{x}_i(t+1))$.
(3) Both $\vec{P}_i(t)$ and $\vec{x}_i(t+1)$ are infeasible, but $viol(\vec{P}_i(t)) > viol(\vec{x}_i(t+1))$

According to feasibility-based rules, there will be many infeasible particles if the feasible region is a highly constrained search space, which will lower the search efficiency. Thus, we define a feasible particle set as following:

$$FPS = \{\vec{P}_i(t) | \vec{P}_i(t) \quad is \quad feasible\}$$

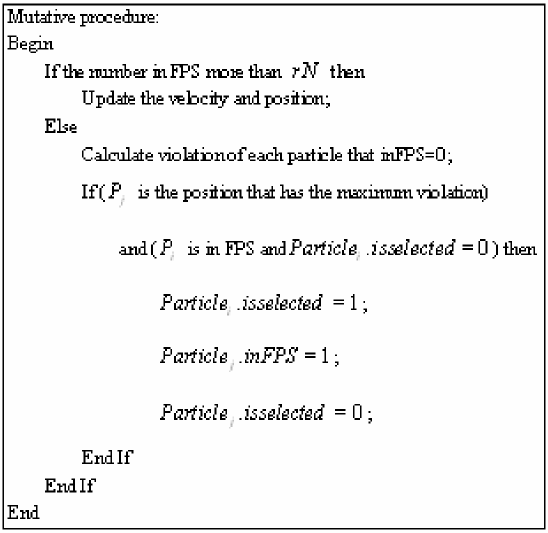
If the particle number in FPS is more than rN (where r is a constant number that less than 1; N is the particle number), update velocity and position of each particle; otherwise, select one P_{best} in FPS to substitute P_{best} that the violation value is maximal. There are many methods to implement the selection, such as stochastic method, sequence method. In IPSO algorithm, sequence method is chosen to do the selection. Figure 2 shows the mutative procedure.

In standard PSO, g_{best} is the best historical position that the entire swarm has passed. In IPSO algorithm, g_{best} is the best P_{best} in FPS that has the best fitness value.



```
Function IPSO Algorithm
Begin
  For each particle
    Initialize position and velocity;
    Compute violation value;
    Compute fitness value;
    $P_{best}$ = the particle's best history position;
    Put $P_{best}$ in FPS if $P_{best}$ is in feasible region;
  EndFor
  Do
    $g_{best}$ = $P_{best}$ that has the best fitness value in the feasible region;
    If the total number of $P_{best}$ in FPS is less than or equal to $rN$ then
        Mutate $P_{best}$ of particle that the violation value is maximal with one $P_{best}$ in FPS;
    EndIf
    Compute the average velocity;
    For each particle
        Calculate the new velocity and position with formula (5) and (6);
        Calculate the fitness value;
        If the fitness value is better than $P_{best}$ according to feasibility-based rules then
            $P_{best}$ =the particle's position;
        EndIf
    EndFor
  While stopping condition not satisfied
End.
```

Fig. 1. Pseudocode of the IPSO algorithm



```
Mutative procedure:
Begin
    If the number in FPS more than $rN$ then
        Update the velocity and position;
    Else
        Calculate violation of each particle that inFPS=0;
        If($P_i$ is the position that has the maximum violation)
            and($P_i$ is in FPS and $Particle_i.isselected = 0$) then
                $Particle_i.isselected = 1$;
                $Particle_i.inFPS = 1$;
                $Particle_i.isselected = 0$;
        End If
    End If
End
```

Fig. 2. The mutative procedure

3.2 Updating Velocity and Position

According to feasibility-based rules, feasible solutions are always considered better than infeasible solutions. That may cause the overpressure of selecting feasible solutions so as to result in premature convergence. So to improve the exploration of the particle, in our algorithm, the swarm is manipulated according to the following update equations:

$$\begin{aligned} v_{id}(t+1) = \omega v_{id}(t) + c_1 r_1 (P_{id}(t) - x_{id}(t)) + \\ c_2 r_2 (P_{gd}(t) - x_{id}(t))) - \omega' \bar{v}_{id}(t) \end{aligned} \tag{5}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{6}$$

Where ω is a parameter called the inertia weight, c_1 and c_2 are positive constants respectively referred as cognitive and social parameters, r_1 and r_2 are random numbers uniformly distributed in [0,1], ω' is a parameter that we call the turbulence parameter ,$\bar{v}_{id}(t)$ is average velocity of the swarm at generation t.

4 Experiment and Discussions

To evaluate the performance of the proposed algorithm, we conducted a series of experiments on the well known Michalewicz' benchmark functions[19] extended by Runarsson and Yao[12]. All the testing problems are described in detail in [12].These test functions selected include characteristics that are representative of what can be considered "difficult" global optimization problems for an evolutionary algorithm.

Problems g02, g03, g08 and g12 are maximization problems and they were converted into minimization problems using $-f(\vec{x})$. Problems g03, g05, g11 and g13 involve equality constraints. All equality constraints $h_j(\vec{x}) = 0, j = 1, 2, \cdots, l$ have been transformed into inequality constraints $|h_j(\vec{x})| \leq \delta$, using the degree of violation $\delta = 10^{-4}$ when the particles were initialed. A total of 40 particles were employed, the maximum number of iterations were set to 8500 per run, and 30 independent runs of our algorithm were executed for each problem. The parameters in update equation are set as follows: ω linearly decreases from 0.9 to 0.4; to improve the diversity of the particle, c_1 is set to decrease from 3.6 to 2 and c_2 is set to increase from 0.2 to 2 respectively; and turbulence parameter ω' linearly decreases from 0.6 to 0.

Table 1 summarizes the experimental results obtained using our algorithm with the above experimental settings, where "Opt" represents the known "optimal" solution for each problem, "Std" stands for "Standard deviation" of the obtained statistics for the 30 independent runs. The statistical results of our algorithm show that IPSO algorithm was able to find the global optimum in almost all test functions except for g10, and the standard deviation is small.

Moreover, we compared our results with respect to three algorithms representative of the state-of-the-art in the area: Stochastic Ranking (SR) [5], the Constraint-Handling Mechanism for PSO (CHMPSO) [13] and the Simple Multimembered Evolution Strategy (SMES) [21]. Our comparison of results with respect to the three previously described is presented in Table 2, Table 3 and Table 4.

Table 1. Experimental results on 13 benchmark functions using IPSO with $I_{max} = 8500$

Pro	Opt	Best	Mean	Worst	Std
g01	-15	-15	-15	-15	0.000000
g02	0.803619	0.803603	0.663532	0.488776	0.015865
g03	1	1.004987	1.004860	1.004309	0.000024
g04	-30665.539	-30665.538672	-30665.538672	-30665.538672	0
g05	5126.4981	5126.498110	5126.500182	5126.507464	0.000460
g06	-6961.81388	-6961.813875	-6961.813856	-6961.813794	0.000003
g07	24.306	24.334560	24.961397	26.011018	0.081870
g08	0.095825	0.095825	0.095825	0.095825	0
g09	680.630	680.630765	680.668078	680.986895	0.0011918
g10	7049.3307	7251.206603	7377.286579	7582.967352	18.443359
g11	0.75	0.749000	0.749001	0.749003	0.000000
g12	1	1.000000	1.000000	1.000000	0.000000
g13	0.0539498	0.060055	0.060056	0.060065	0.000000

When comparing IPSO with respect to SR, we can see that IPSO found better solutions for g03, g06, g11 and similar result in g01, g04, g08 and g12. For problem g02, IPSO found better solution in terms of better solution. For problems g05 and g10, though the results of IPSO in terms of best solution are worse than SR, it does better in terms of mean solutions and worst solutions, which indicates that the capability to find optimal solution of IPSO algorithm is better than SR's for these two problems.

Table 2. Comparison of our IPSO with respect to SR[21]

Problem	Optimal	Best Result		Mean Result		Worst Result	
		IPSO	SR	IPSO	SR	IPSO	SR
g01	-15	-15	-15	-15	-15	-15	-15
g02	0.803619	0.803603	0.803515	0.663532	0.781975	0.488776	0.726288
g03	1	1.004987	1.000000	1.004860	1.000000	1.004309	1.000000
g04	-30665.539	-30665.538672	-30665.539	-30665.538672	-30665.539	-30665.538672	-30665.539
g05	5126.4981	5126.498110	5126.497	5126.500182	5128.881	5126.507464	5142.472
g06	-6961.81388	-6961.813875	-6961.814	-6961.813856	-6875.94	-6961.813794	-6350.262
g07	24.306	24.334560	24.07	24.961397	24.374	26.011018	24.642
g08	0.095825	0.095825	0.095825	0.095825	0.095825	0.095825	0.095825
g09	680.630	680.630765	680.630	680.668078	680.656	680.986895	680.763
g10	7049.3307	7251.206603	7054.316	7377.286579	7559.192	7582.967352	8835.655
g11	0.75	0.749000	0.75	0.749001	0.75	0.749003	0.75
g12	1	1.000000	1	1.000000	1	1.000000	1
g13	0.0539498	0.060055	0.053957	0.060056	0.057006	0.060065	0.216915

Table 3. Comparison of our IPSO with respect to CHMPSO[13]

Problem	Optimal	Best Result		Mean Result		Worst Result	
		IPSO	CHMPSO	IPSO	CHMPSO	IPSO	CHMPSO
g01	-15	-15	-15	-15	-15	-15	-15
g02	0.803619	0.803603	0.803432	0.663532	0.790406	0.488776	0.750393
g03	1	1.004987	1.004720	1.004860	1.003814	1.004309	1.002490
g04	-30665.539	-30665.538672	-30665.5	-30665.538672	-30665.5	-30665.538672	-30665.5
g05	5126.4981	5126.498110	5126.64	5126.500182	5461.081333	5126.507464	6104.75
g06	-6961.81388	-6961.813875	-6961.81	-6961.813856	-6961.81	-6961.813794	-6961.81
g07	24.306	24.334560	24.3511	24.961397	25.355771	26.011018	27.3168
g08	0.095825	0.095825	0.095825	0.095825	0.095825	0.095825	0.095825
g09	680.630	680.630765	680.638	680.668078	680.852393	680.986895	681.553
g10	7049.3307	7251.206603	7057.59	7377.286579	7560.047857	7582.967352	8104.31
g11	0.75	0.749000	0.749999	0.749001	0.750107	0.749003	0.752885
g12	1	1.000000	1	1.000000	1	1.000000	1
g13	0.0539498	0.060055	0.068665	0.060056	1.716426	0.060065	13.6695

Compared with respect to CHMPSO, IPSO algorithm found better or similar solutions for all problems except g10. For problem g10, IPSO algorithm got better solutions in terms of mean solution and worst solution.

When comparing against SMES, IPSO found better solutions for g03, g05, g06, g11 and g13, and the same or similar results in g04, g08 and g12. For problems g02, g07 and g09, IPSO algorithm found better solutions in terms of best result.

Moreover, the computational cost measured in the number of evaluations of the objective function (FFE) performed by IPSO algorithm is 340,000 FFE, lower than the Stochastic Ranking (SR) which performed 350,000 FFE, and the same as CHMPSO and SMES.

Table 4. Comparison of our IPSO with respect to SMES[22]

Problem	Optimal	Best Result		Mean Result		Worst Result	
		IPSO	SMES	IPSO	SMES	IPSO	SMES
g01	-15	-15	-15	-15	-15	-15	-15
g02	0.803619	0.803603	0.803601	0.663532	0.785238	0.488776	0.751322
g03	1	1.004987	1.000	1.004860	1.000	1.004309	1.000
g04	-30665.539	-30665.538672	-30665.539	-30665.538672	-30665.539	-30665.538672	-30665.539
g05	5126.4981	5126.498110	5126.599	5126.500182	5174.492	5126.507464	5304.167
g06	-6961.81388	-6961.813875	-6961.81	-6961.813856	-6961.284	-6961.813794	-6952.482
g07	24.306	24.334560	24.3511	24.961397	24.4751	26.011018	24.843
g08	0.095825	0.095825	0.095825	0.095825	0.095825	0.095825	0.095825
g09	680.630	680.630765	680.638	680.668078	680.643	680.986895	680.719
g10	7049.3307	7251.206603	7057.59	7377.286579	7253.047	7582.967352	7638.366
g11	0.75	0.749000	0.749999	0.749001	0.75	0.749003	0.75
g12	1	1.000000	1	1.000000	1.000	1.000000	1
g13	0.0539498	0.060055	0.068665	0.060056	0.166385	0.060065	0.468294

5 Conclusions and Future Work

This paper presents an improved particle swarm optimization (IPSO) to solve constrained optimization problems, which handles constraints based on certain feasibility-based rules. IPSO introduces the turbulence operator to improve the swarm diversity and to overcome the premature convergence validly. The record about the feasible P_{best} in FPS and the mutation operator on the P_{best} with a maximal violation value can lead the swarm flying to the feasible region quickly. Obviously, the principle of IPSO algorithm is simple and its implement is easy, the experimental results also show the technique is highly competitive.

The future work is to study alternative mechanisms to accelerate convergence while keeping the same quality of the results achieved in this paper.

References

1. Coello Coello, C.A.: Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. Comput. Meth. Appl. Mech. Eng. 191, 1245–1287 (2002)
2. Michalewicz, Z.: A survey of constraint handling techniques in evolutionary computation methods. In: Mcdonnell, J.R., Reynolds, R.G., Fogel, D.B. (eds.) Proceedings of the 4th Annual Conference on Evolutionary Programming, pp. 135–155. MIT Press, San Diego (1995)
3. Coello Coello, C.A.: Use of a self-adaptive penalty approach for engineering optimization problems. Comput. Ind. 41, 113–127 (2000)
4. Koziel, S., Michalewicz, Z.: Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization. Evol. Comput. 7(1), 19–44 (1999)
5. Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. IEEE Trans. Evol. Comput. 4(3), 284–294 (2000)
6. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)

7. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39–43 (1995)
8. Floudas, C.A., Pardalos, P.M.: A Collection of Test Problems for Constrained Global Optimization Algorithms. LNCS, vol. 455. Springer, Heidelberg (1990)
9. Himmelblau, D.M.: Applied Nonlinear Programming. McGraw-Hill, New York (1972)
10. Parsopoulos, K.E., Vrahatis, M.N.: Particle swarm optimization method for constrained optimization problems. In: Proceedings of the Euro-International Symposium on Computational Intelligence (E-ISCI 2002), Slovakia (2002)
11. Lu, H.Y., Chen, W.Q.: Self-adaptive velocity particle swarm optimization for solving constrained optimization problems. Journal of Global Optimization 41(3), 427–445 (2008)
12. Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. IEEE Trans. Evol. Comput. 7, 19–44 (1999)
13. Toscano Pulido, G., Coello Coello, C.A.: A constraint-handling mechanism for particle swarm optimization. In: Proceedings of the 2004 Congress on Evolutionary Computation, vol. 2, pp. 1396–1403 (2004)
14. He, Q., Wang, L.: A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Mathematics and Computation 186, 1407−1422 (2007)
15. Hu, X., Eberhart, R.C.: Solving constrained nonlinear optimization problems with particle swarm optimization. In: Proceedings of 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002), Orlando, USA (2002)
16. Zhang, W.J., Xie, X.F.: DEPSO: hybrid particle swarm with differential evolution operator. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics (October 2000); IEEE Trans. Evol. Comput. 4, 284–294 (2000)
17. Shi, Y.H., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings IEEE International Conference on Evolutionary Computation, Anchorage, pp. 69–73 (1998)
18. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm Intelligence. Morgan Kaufman Publishers, San Francisco (2001)
19. Michalewicz, Z., Schoenauer, M.: Evolutionary algorithms for constrained parameter optimization problems. Evol. Comput. 4, 1–32 (1996)
20. Muñoz Zavala, A.E., Hernández Aguirre, A., Villa Diharce, E.R.: Constrained optimization via particle evolutionary swarm optimization algorithm (PESO). In: GECCO 2005, Washington, DC, USA, pp. 209–216 (2005)
21. Efrén, M.M., Coello Coello, C.A.: A simple multimembered evolution strategy to solve constrained optimization problems. Evolutionary Computation 4(1), 1–32 (1996)
22. Chang, J.F., Chu, S.C., Roddick, J.F., Pan, J.S.: A Parallel Particle Swarm Optimization Algorithm with Communication Strategies. Journal of Information Science and Engineering 21(4), 809–818 (2005)
23. Chu, S.C., Pan, J.S.: Intelligent Parallel Particle Swarm Optimization Algorithms. In: Nedjah, N., Alba, E., Macedo Mourelle, L. (eds.) Studies in Computational Intelligence Series, vol. 22, pp. 159–175. Springer, Heidelberg (2006)
24. Chu, S.C., Tsai, P.W., Pan, J.S.: Parallel Particle Swarm Optimization Algorithms with Adaptive Simulated Annealing. In: Abraham, A., Grosan, C., Ramos, V. (eds.) Studies in Computational Intelligence Series, vol. 31, pp. 261–279. Springer, Heidelberg (2006)

A Trajectory-Based Point Tracker Using Chaos Evolutionary Programming

Shu-Mei Guo[1], Chih-Yuan Hsu[1], Po-Nung Wu[1], and Jason Sheng-Hong Tsai[2]

[1] Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, 701, Taiwan, R.O.C.
{guosm,P7893125,P7694105}@mail.ncku.edu.tw
[2] Department of Electrical Engineering
National Cheng Kung University
Tainan, 701, Taiwan, R.O.C.
shtsai@mail.ncku.edu.tw

Abstract. A trajectory-based point tracker using chaos evolutionary programming (CEP) algorithm is proposed in this paper. While motion constraints such as rigidity and small motion which are imposed by previous approaches are liberated, the proposed CEP is proved to be effective for establishing point correspondence between two consecutive frames sampled at a fixed interval. The whole point trajectory within the sample interval is then reconstructed by polynomial interpolation. Our experimental results demonstrate that the proposed point tracker can accurately locate target under different kinds of situations like object deformation, occlusion, and sudden motion as well.

Keywords: Point tracking, Chaos, Evolutionary programming, Bilinear interpolation, Polynomial interpolation.

1 Introduction

Point tracking is a very challenging task under many applications such as motion estimation [1], robot vision control [2], video database management, surveillance, or three-dimensional (3-D) reconstruction [3]. The first step of point tracking is to build up feature point correspondence over video sequence. A primary problem encountered in the correspondence is that it has been previously proved to be combinatorially explosive considering the amount of all the possible trajectory sets which need to be explored to find the correct trajectory set even for a small-size frame sequence [4].

Some approaches have been proposed to handle the combinatorial complexity based on some constraints on point motion, such as rigidity, large displacement, small position change, and common motion. Mehrotra [5] proposed an approach employing motion uniformity constraint, which assumes that the trajectory of a point over a short period is as “short” and “smooth” as possible. Its advantage is that it permits point trajectories to terminate and start at intermediate frames of the image sequence. Salari and Sethi [6] later proposed an improved algorithm by introducing

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 212–220, 2009.

the notion of phantom feature points and solved the missing and spurious measurement problem. Notwithstanding, its iterative refinement process is computationally expensive. Sethi and Jain [7] proposed two different algorithms, GE (Greedy Exchange) and MGE (Modified Greedy Exchange). Disregarding their limitations on handling motion irregularities, GE and MGE also suffer from their slow computation speed. Wang [8] proposed a genetic algorithm-based approach to feature point correspondence across image sequence to deal with occluded points and outliers as well as to improve computational efficiency.

In order to circumvent motion constraints aforementioned, the chaos evolutionary programming (CEP) algorithm is proposed to solve the point correspondence problem more efficiently. With the help of CEP, a more sophisticated motion model which incorporates parameters such as displacement, in-plane rotation angle and object scaling can be constructed. Additionally, to further improve computational efficiency, while CEP is applied to a selective number of frames to locate the target point, polynomial interpolation is employed to reconstruct the tracking trajectory in-between. Fig. 1 illustrates the block diagram of the proposed tracker.

The organization of the remainder of this paper is as follows. Section 2 formulates the optimization problem for the CEP algorithm, which is to be discussed in Section 3. Next, the proposed point tracker is introduced in Section 4 and the experimental results are given in Section 5. Finally, Section 6 is our conclusions.

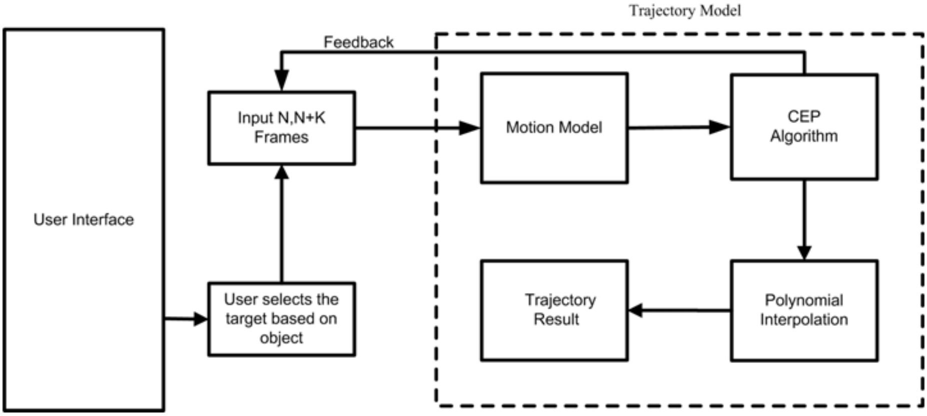


Fig. 1. The tracker scheme

2 Problem Formulation

2.1 Motion Model

For a target point based on its window, define the state vector at time as

$$\tilde{x}_{i,j} = \left[x(t_k)_{i,j} \;\; y(t_k)_{i,j} \;\; \theta(t_k) \;\; s(t_k)\right]^T, \;\; i, j \in W : windows, \qquad (1)$$

where $\left(x(t_k)_{i,j} \;\; y(t_k)_{i,j}\right)$ is the position of the corresponding point in the k-th frame, and $\theta(t_k)$ is the rotation angle and $s(t_k)$ is the scaling factor of the k-th image with respect to the reference frame, which, in our experiment, is the 0-th image.

For a point of interest, an affine transformation to describe the target motion between two windows in which the matching points are likely to appear can be expressed by the following equation:

$$\tilde{x}_{i,j}(t_k) = \begin{bmatrix} \begin{bmatrix} x(t_0)_{origin} \\ y(t_0)_{origin} \end{bmatrix} + s(t_k) \begin{bmatrix} \cos\theta(t_k) & -\sin\theta(t_k) \\ \sin\theta(t_k) & \cos\theta(t_k) \end{bmatrix} \begin{bmatrix} x(t_0)_{i,j} - x(t_0)_{origin} \\ y(t_0)_{i,j} - y(t_0)_{origin} \end{bmatrix} + \begin{bmatrix} \Delta x(t_k) \\ \Delta y(t_k) \end{bmatrix} \\ \theta(t_k) \\ s(t_k) \\ \end{bmatrix}, \tag{2}$$

where $(x(t_0)_{origin}, y(t_0)_{origin})$ is the left-upper point of the window in the reference frame, and $\Delta x(t_k), \Delta y(t_k), \theta(t_k)$, and $s(t_k)$ are the translational, rotational, and scaling parameters of the k-th frame, which is illustrated in Fig. 2.

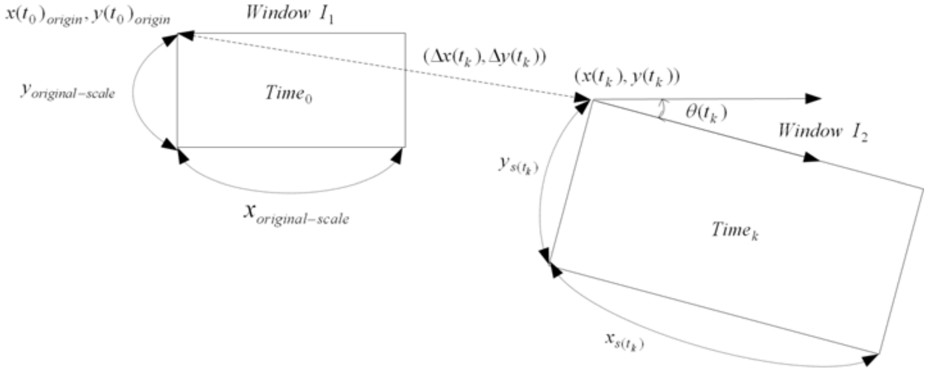


Fig. 2. Motion model

2.2 Feature Matching

For each set of motion parameters, the following procedure is performed to measure the similarity between the candidate and the target point. First, identify the candidate window from the k -th frame. We refer to the window from the reference frame as the reference window I_1; the other one is called the candidate window I_2 as showed in Fig. 2.

Second, according to the scaling parameter of window I_2, bilinear interpolation is used to perform the reduction or enlargement of I_2. Then, the difference between I_1 and I_2 is calculated according to Eq. (3).

$$SAD = \sum_{(x,y)\in W} abs\left[I_k(x,y) - I_0(x,y)\right], \tag{3}$$

where W : Window of target.

2.3 Polynomial Interpolation

Polynomial interpolation is to interpolate a given data set by a polynomial function. After having coefficients of the polynomial equation, this equation can be used to interpolate other points.

2.3.1 Application to Construct Point Trajectory

The proposed tracker is hoped to use CEP to build up the point correspondence with a selective number of frames so that the computational load can be further reduced. Therefore, the whole point trajectory is to be reconstructed by the polynomial interpolation. Suppose that the polynomial interpolation is as the form

$$p(x) = \sum_{m=0}^{n} a_m x^m = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0. \tag{4}$$

The statement that p interpolates the date points means that $p(x_i) = y_i, i = 0,\ldots,n$. Substituting these constraints into Eq. (4) yields a system of linear equations in the coefficients $a_k, k = 0,\ldots,n$. In this paper, a third order interpolation equation is used to construct the point trajectory.

However, the interpolated points may be unreasonable. And, the results display that is not what we expected. In order to solve this problem, it can modify the interpolated points by Eqs. (5)-(7).

$$p(i) = \max\{p(n), p(n+r)\} \quad \text{if } p(i) > \max\{p(n), p(n+r)\}, \tag{5}$$

$$p(i) = \min\{p(n), p(n+r)\} \quad \text{if } p(i) < \min\{p(n), p(n+r)\}, \tag{6}$$

$$p(i) \text{ remains invariant, otherwise,} \tag{7}$$

where $p(n)$ and $p(n+r)$ are the original points, and the points of $p(i)$ at $i = n+1, n+2, \cdots, n+r-1$ are interpolated points.

3 CEP Algorithm

Evolutionary programming (EP) [9], [10] is similar to Genetic algorithm (GA). It is a parallel optimization and computational technique. However, it's possible that solution generated by EP be very close to but not the actual optimal value. As a result, we adopt a modified chaos optimal algorithm (COA) [11], [12] to search around the best result produced from EP. There will be more detailed discussion as follows.

3.1 The Chaos Optimization Algorithm

The chaotic equation for COA can be selected as the logistic mapping, namely,

$$t_{k+1} = f(\mu, t_k) = \mu t_k (1 - t_k), \quad k = 1,2,\ldots,N, \tag{8}$$

where μ is a control parameter. It is easy to find that Eq. (8) becomes chaotic state as $\mu = 4$. When $\mu = 4$, the equation is described as

$$t_{k+1} = 4t_k(1 - t_k). \tag{9}$$

The chaotic space of Eq. (9) is [0 1]. The system is easy to be affected by initial conditions. A slight change of initial conditions will result in a chaotic motion during the long-time behavior of a chaotic system. Besides, the chaotic motion can go non-repeatedly through every state in a certain domain.

3.2 The Chaos Evolutionary Programming Algorithm

If the optimization problems are continuous problems and the constraints of the variables are known, the optimization problems can be described as

$$\min \mathit{f(x_i\,)}\ \mathit{or}\ \max \mathit{f(x_i\,)}\quad i = 1,2,...,n,\quad \mathit{for}\ a_i \le x_i \le b_i. \tag{10}$$

Suppose that the natural numbers are represented in the scale of notation with radix *R*, so that

$$n = a_0 + a_1 R + a_2 R^2 + \cdots + a_m R^m,\quad 0 \le a_i \le R. \tag{11}$$

Write the digits of these numbers in the reverse order, preceded by a decimal point. This gives the number

$$\phi_R(n) = a_0 R^{-1} + a_1 R^{-2} + \cdots + a_m R^{-m-1}. \tag{12}$$

Halton [13] extended the two-dimensional result of Van Der Corput [14] to ρ-dimensions, where $R_1, R_2, ..., R_p$ are mutually coprime.

Suppose we have *n* populations and each population has *m* components. Since $\phi_R(n) < 1$, this range has to be scaled to satisfy the parameter. Let the interval real ($\Im\Re$) matrix $X \in \Im\Re^{n\times m}$ be a set of degenerate real matrices defined by

$$X = [L, U] = \{[x_{ij}]\, l_{ij} \le xi_j \le u_{ij}; 1 \le i \le n, 1 \le j \le m\}, \tag{13}$$

where L and U are constant real matrices. We introduce the variable $\eta_{ij}, 0 \le \eta_{ij} \le 1$ such that

$$x_{ij} = l_{ij} + \eta_{ij}(u_{ij} - l_{ij}), \tag{14}$$

and use the notation $\eta = [\eta_{11}, \cdots, \eta_{1m}, \eta_{21}, \cdots, \eta_{2m}, \eta_{n1}, \cdots, \eta_{nm}]$. Then the interval matrix *X* can be denoted as $X(\eta)$. Let $\eta_{11} = \phi_2(n)$ 、 $\eta_{12} = \phi_3(n)$ 、 $\eta_{13} = \phi_5(n) \cdots$, and so on, to construct the desired initial population of size *N* for $n = 1, 2, \cdots, N$. Then CEP by merging of the EP and COA can be used to optimize these problems. Total Steps are individual population, objective function, fitness function, probability function, mutation, selection, penalty, EP termination condition, and COA search, respectively. Please refer to [15] for more detail.

4 The Proposed Point Tracker

The proposed point tracker consists of two stages. The first stage is building point correspondence by CEP, which also means to find out the best match denoting by motion parameters $\Delta x(t_k), \Delta y(t_k), \theta(t_k)\ and\ s(t_k)$ as described in Section 2.1. Point correspondence is established on a selective number of frames, which are sampled at a fixed interval. The second stage is to reconstruct the whole point trajectory by polynomial interpolation on the results from the first stage.

4.1 Point Correspondence by CEP

CEP starts with a population of individuals, each of which represents a possible match for the target point. Each individual has ρ numbers of genes q. Here $\rho = 4$ and each gene stands for the four motion parameters $\Delta x(t_k), \Delta y(t_k), \theta(t_k)\ \ and\ s(t_k)$ defined in Section 2.1. The objective function score for each individual is defined as Eq. (3). Therefore, the lower the objective score, the better. We form the initial population $P_0 = [P_1, P_2, \cdots, P_N]$ of size N according to quasi random sequence (QRS). After the first two iterations, assuming that the target motion is usually smooth, the best individuals from the previous two iterations can also provide useful information for locating the best match in the following frames. Thus, we use the rolling average of the translational displacement from the last two iterations as the initial translational displacement for a small group of individuals in the next iteration.

4.2 Dealing with Irregularities

When occlusion or sudden motion occurs, then the sampling interval is halved as shown in Eq. (15)

$$\mathrm{int}\,erval_{\mathrm{mod}\,ified} = \mathrm{int}\,erval_{original} / 2, \tag{15}$$

where interval is larger then 1, and only takes the integer.

Once the target is located, the value of interval is restored to the original value. For instance, if the interval is ten and occlusion happens at 110-th frame, the proposed tracker will try to locate the given target at 115-th frame. If point matching still fails at 115-th frame, the interval value remains at five.

On the other hand, if the interval is six, the tracker will trace back the given target at 51-th frame when speeding suddenly happens at 54-th frame for instance. However, if point matching still fails in 51-th frame, interval is halved to 1.

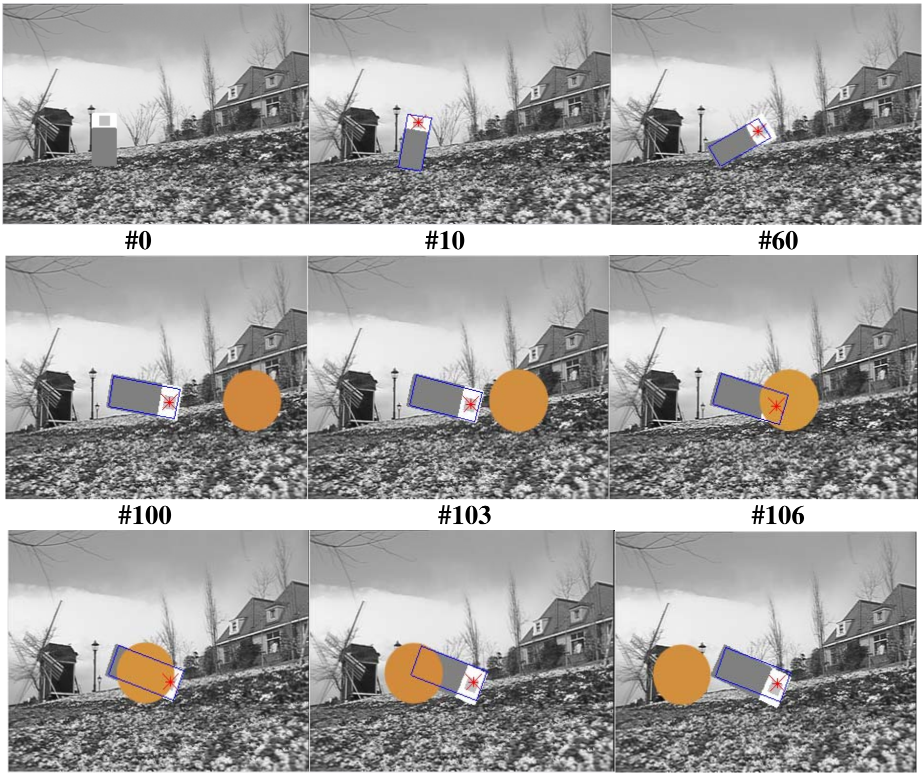
About applications of polynomial interpolation, when unusual motion conditions such as occlusion and sudden motion happen, the original sample point at a given interpolate interval will be replaced accordingly by the one where the target point is tracked. Therefore one interpolates the appropriate points by those records which are the tracked points and their corresponding frame locations.

5 Experimental Results

Four different video sequences **Virtual** and **Boot,** are used in our experiment. Since each video sequence demonstrates different types of motion complexity, to speed up the process, the search range of motion parameters $\Delta x(t_k), \Delta y(t_k), \theta(t_k),$ and $s(t_k)$ for the CEP algorithm is set separately, which is given in Table 1.

Table 1. The search range of parameters and the settings used in the experiments

	Virtual	Boot
$\Delta x(t_k), \Delta y(t_k)$	(+13 ~ −13)	(+25 ~ −25)
$\theta(t_k)$	$(+15^\circ \sim -15^\circ)$	$(+3^\circ \sim -3^\circ)$
$s(t_k)$	(0.94 ~ −1.06)	(0.94 ~ 1.06)
Interval	10	6
Occlusion-key	1	0
Threshold	35,000	35,000



#0 #10 #60

#100 #103 #106

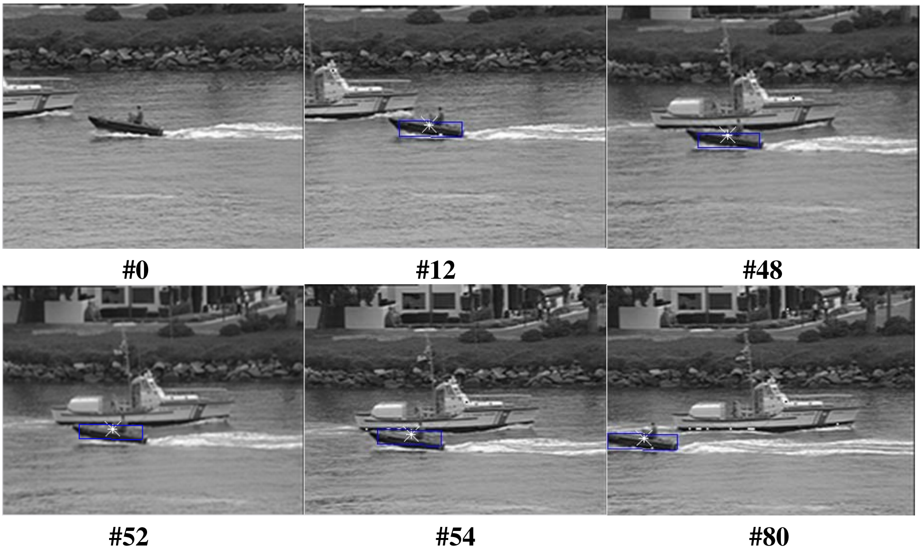
#110 #113 #115

Fig. 3. Trajectory of the point in the sequence "**Virtual**"

In the following figures, the tracked point is marked by "✳" and the rectangular box defines the object as the window of target. First, the tracking result of the sequence **Virtual** using **Garden** as background is illustrated in Fig. 3. The original size of the target object is 24×56.

The sequence Virtual demonstrates several motion conditions such as rotation, translation, enlargement, and occlusion. Frames #0 to #60: rotation, translation, and enlargement; frames #100 to #115: occlusion. Frames #103, #106, #110, and #113 are results from polynomial interpolation.

The second sequence, **Boot,** is a 80-frame (320×240) sequence. The tracking results are shown in Fig. 4. The original size of the target object is 64×15.



#0 **#12** **#48**

#52 **#54** **#80**

Fig. 4. Trajectory of the point in the sequence "**Boot**"

Frames #48 to #55: speeding happens. Frames #52 and #54 are results from polynomial interpolation**.**

6 Conclusions

In this paper, a trajectory-based point tracker using CEP is proposed. The CEP algorithm is used to establish point correspondence efficiently. With the help of polynomial interpolation, the whole point trajectory can be reconstructed so that the computational efficiency of the proposed point tracker can be further improved. The experimental results show that the proposed point tracker can track points accurately and handle issues such as object deformation and motion irregularities effectively.

Acknowledgements. This work was supported by the National Science Council of Republic of China under contracts NSC 96-2221-E-006-292-MY3 and NSC 96-2221-E-006-302.

References

1. Wang, Y., Baciu, G.: Human motion estimation from monocular image sequence based on cross-entropy regularization. Pattern Recognition Letters 24, 315–325 (2003)
2. Mezouar, Y., Chaumette, F.: Model-free optimal trajectories in the image space: application to robot vision control. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1155–1162 (2001)
3. Sturm, P.: Structure and motion for dynamic scenes - the case of points moving in planes. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2351, pp. 867–882. Springer, Heidelberg (2002)
4. Rangarajan, K., Shah, M.: Establishing motion correspondence. CVGIP: Image Understanding 54, 56–73 (1991)
5. Mehrotra, R.: Establishing motion-based feature point correspondence. Pattern Recognition 31, 23–30 (1998)
6. Salari, V., Sethi, I.K.: Feature point correspondence in the presence of occlusion. IEEE Trans. Pattern Anal. Mach. Intell. 12, 87–91 (1990)
7. Sethi, I.K., Jain, R.: Finding trajectories of feature points in a monocular image sequence. IEEE Trans. Pattern Anal. Mach. Intell. 9, 56–73 (1987)
8. Wang, Y.: Feature point correspondence between consecutive frames based on genetic algorithm. International Journal of Robotics and Automation 21, 35–38 (2006)
9. Fogel, D.B.: Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. IEEE Press, Piscataway (1999)
10. Guo, S.M., Shieh, L.S., Chen, G., Coleman, N.P.: Observer-type kalman innovation filter for uncertain linear systems. IEEE Trans. Aerospace Elec. Syst. 37, 1406–1418 (2001)
11. Li, B., Jiang, W.S.: Optimizing complex functions by chaos search. Cybernetics and Systems 29, 409–419 (1998)
12. Yan, X.F., Chen, D.Z., Hu, S.X.: Chaos-genetic algorithms for optimizing the operating conditions based on RBF-PLS model. Computers and Chemical Engineering 27, 1393–1404 (2003)
13. Halton, J.H.: On the efficiency of certain quasi-random sequences of points in evaluating multidimensional Integrals. Numerische Mathematik 2, 84–90 (1960)
14. Van Der Corput, J.C.: Verteilungsfunktionen. Proc. Kon. Akad. Wet. Amsterdam 38, 1058–1066 (1935)
15. Guo, S.M., Liu, K.T., Tsai, J.S.H., Shieh, L.S.: An observer-based tracker for hybrid interval chaotic systems with saturating inputs: The chaos-evolutionary-programming approach. Computers and Mathematics with Applications 55, 1225–1249 (2008)

Caching in the TSP Search Space

David Karhi and Dan E. Tamir

Department of Computer Science
Texas State University
San Marcos Texas, 78666 USA
{dk1037,dt19}@txstate.edu

Abstract. Heuristic search techniques can often benefit from record keeping and saving of intermediate results, thereby improving performance through exploitation of time / space tradeoffs. Iterative hill climbing (ITHC) is one of these heuristics. This paper demonstrates that record keeping in the ITHC domain can significantly speed up the ITHC. The record keeping method is similar to the mechanism of a cache. The new approach is implemented and tested in the traveling salesperson search space. The research compares a traditional random restart (RR) procedure to a new greedy enumeration (GE) method. GE produces Hamiltonian-cycles that are about 10% shorter than the RR. Moreover, the cached RR achieves a speedup of 3x with a relatively small number of cities and only 20% with a medium number of cities (~17). The cached GE shows a highly significant speedup of 4x over traditional methods even with a relatively large number of cities (>80).

Keywords: Heuristic search, Traveling salesperson, Hill climbing, Iterative hill climbing, Cache.

1 Introduction

Artificial intelligence (AI) applications are often exponential with respect to the problem size and require NP complete or intractable procedures for finding an optimal solution [1]. A common approach for solving such problems is to define a search space that is an enumeration of potential solutions to the problem or steps toward the solution, and perform heuristic search in that space [2]. The heuristics prunes the search space and reduces complexity by considering only a subset of the possible solutions. It looks for the globally optimal solution but, due to the complexity of the problem and the limited amount of resources, may settle for a locally optimal solution. A multitude of heuristic search and combinatorial optimization methods is under extensive research. The list of these techniques includes the A*, Hill Climbing, iterative hill climbing (ITHC), genetic algorithms, taboo search, swarm, and simulated annealing (SA) [3, 4, 5, 6].

Despite the reduction in problem size, heuristic search algorithms are often very complex and require careful utilization of computer resources such as processing power and storage space. Many AI heuristic search algorithms are designed for optimal utilization of processing power in order to reduce processing time and do not

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 221–230, 2009.

fully exploit the storage space or the trade-off between time and space complexity. Several researchers have studied time/space trade-offs. Hertel and Pitassi, studied time/space trade-offs within the context of satisfiability algorithms [7]. They found that the long-held assumption that there is a nearly linear trade-off between time and space is not always true within the domain of satisfiability algorithms, and at times, the trade-off could become exponential. Allen and Darwiche studied time/space trade-offs within the domain of probabilistic inference [8]. They found that time requirements could be significantly reduced through record keeping. This optimization made exact inference possible in situations that were previously impractical. They referred to their method as caching in probabilistic reasoning. Nevertheless, their cache is static; they do not consider cache replacement policies, or cache organization issues. Moreover, their findings do not pertain to ITHC. To the best of our knowledge, this is the first study of time/space trade-offs within the context of ITHC.

ITHC is a primitive type of heuristic. Nevertheless, in many cases it is the only viable solution to several "hard" optimization problems. One of these problems, which occur in the field of Phylogenetics (the biological study of evolutionary relatedness among different groups of organisms), is the problem of finding the minimal parsimony tree [9]. Along with bioinformatics researchers, we are engaged with one of the main problems of Phylogenetics. That is, the problem of finding the Maximum Parsimony Tree (MPT), which is also called the Hamming Distance Steiner Tree Problem. Finding the MPT is an NP-complete problem [9]. Due to its enormous complexity, MPT is typically solved with ITHC. In some cases, "billions and billions" of iterations of hill climbing are applied. Every iteration of the ITHC returns a local optimum. After a large sequence of hill climbing iterations, the biologist pick the best solution encountered so far and considers it as "the solution." There are cases where researchers were forced to run the algorithm on massively parallel computing systems and wait for prolonged time (sometimes more than a year!) for the ITHC to return an acceptable solution that may or may not be optimal. Ganapathy, et al. improved this situation by enhancing the search techniques used with ITHC in MPT, thus reducing the computation time needed. They did not consider adding a cache to further reduce the amount of computation time needed [9].

Originally, our research was concerned with adding a caching mechanism to the ITHC version of the MPT search in an attempt to explore locality of reference. Currently, and as a first phase in the study of ways to improve the MPT, we decided to explore ways to improve the TSP problem by exploitation of the locality of reference. The TSP was chosen as a platform to study time/space trade-offs in ITHC for several reasons. First, the TSP is a well-known problem that has been thoroughly researched and provides ample complexity with which to gage our results [10, 11]. Second, although TSP is incredibly complex, it is easy to analyze and understand. Third, TSP is a problem used in many different fields of research, and is known to many members of the scientific community. Finally, TSP translates well to MPT while being more widely understood than MPT. Our experiments do not limit the TSP to planar graphs, nor do they assume that the triangle inequality holds. Hence, they are general. In the context of this research, the TSP problem is stated in the following way: "Given a complete weighted graph, find the minimal Hamiltonian spanning cycle of the graph."

1.1 Hill Climbing

Hill climbing is a steepest descent greedy search algorithm [6]. There are two parts to the hill climbing algorithm. First, a valid solution, called an initial configuration, is generated. Next, the hill climbing algorithm attempts to improve the current solution by making local changes to the configuration and taking the best new solution that exists in the local search space. Each improvement is referred to as a step. This improvement process continues until hill climbing can no longer find a better solution, at which point the algorithm returns the last solution found. Selecting the best immediate solution at any given step may result in getting trapped at local minima [6]. Regardless, hill climbing is widely used because it is easy to implement. It requires a very small amount of memory and returns relatively good solutions.

1.2 Iterative Hill Climbing

Iterative Hill Climbing (ITHC) is a variation of hill climbing that addresses the problem of getting trapped at a local minimum. In ITHC, once a local minimum is found, another initial configuration is generated from the global search space and the climbing process restarts. This process of generating initial configurations and improving them continues iteratively until the global search space is exhausted or a desired number of iterations are completed. Hence, ITHC can find numerous local minimum, solutions and ideally, one of those minimums is the global minimum.

1.3 Rendezvous

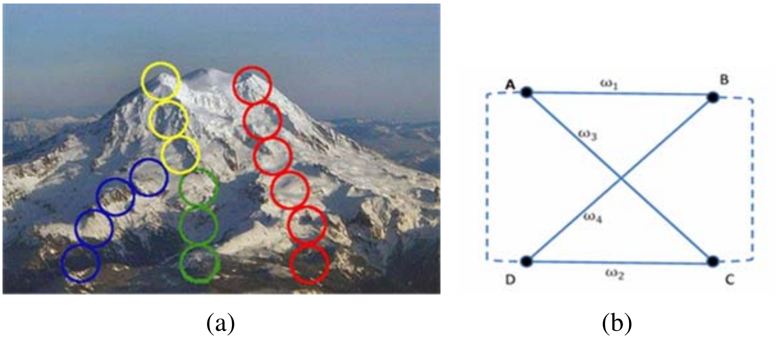
One inherent problem of ITHC is that two random initialization can yield the same configuration. Due to the steepest descent properties of the hill climbing search routine, the two climbers will continue on the same route thereby introducing redundancy. This is depicted in Figure 1a, where the red (left most), blue (middle), and green (right most) circles denote the steps toward solution taken by three climbers. The blue and green climbers meet at a specific point and from this point and on, continue in a single route to the same local minimum. Note that the scenario of employing two climbers to start at the same point is not very likely to occur; a more likely case is that two or more climbers are meeting at the same point. In both cases, only the first climber should be considered and the rest of the climbers are redundant.

This brings an interesting question, which is not sufficiently addressed in the literature. The question relates to the type of records that should be kept throughout the ITHC session and the best policy to save these records in order to minimize rendezvous redundancy. An additional problem with numerous heuristic search methods arises if they have to run to completion before returning a useful solution. This is a concern with algorithms such as MPT as well as with several real time AI applications where the search might be prompted for the "best solution found so far" in the middle of execution. Anytime algorithms produce results with monotonically increasing quality based on the allocated computation time [12, 13, 14]. Similarly, anyspace algorithms produce results with monotonically increasing quality according to the amount of allocated storage space [13, 14]. An anytime and anyspace algorithm exploits time space tradeoffs and monotonically improves the result's quality based on time and space constraints. Iterative hill climbing is an anytime algorithm. Along with the record keeping, it becomes an anytime anyspace algorithm.

2 Hill Climbing / Iterative Hill Climbing in the TSP Search Space

Several heuristics are studied in the context of TSP. Among them are the 2-Opt, 3-Opt, N-Opt, and the Lin-Kernighan (LK) algorithms [10, 11]. The 2-Opt algorithm removes 2 edges from a Hamiltonian spanning cycle and replaces them with 2 edges whose sum is less than the sum of the edges that were removed, while maintaining the Hamiltonian property of the spanning cycle [10, 11]. In this paper, each of these exchanges is called a butterfly. Figure 1b shows an example of a butterfly. In this figure, solid lines denote edges and dotted lines denote a route. Assuming that (ω1+ω2) > (ω3+ ω4), the 2-Opt algorithm would remove ω1 and ω2 and replace them with ω3 and ω4. In addition, this research employs a restricted version of the butterfly where the vertices that are considered for edge exchange (*A* and *B* as well as *C* and *D*) are pairs of consecutive neighbors in the current cycle.

The 2-Opt and 3-Opt are special cases of the N-Opt algorithm, where N edges are swapped at each step. Studies have shown that the quality of the results returned by N-Opt increase as N increases [10, 11]. However, as N increases the computation time required increases as well. It is generally accepted that the increase in computational complexity between 3-Opt and 4-Opt is too large to justify the slight increase in the quality of the results that are returned. Therefore, in practice, N is rarely set higher than 3. The Lin-Kernigha algorithm is similar to N-Opt, except that it dynamically determines which value of N to use at each step. Though LK has been shown to return better results than any of the other heuristics mentioned, it should be noted that LK is an extremely difficult to implement. For this reason, LK is rarely used [10, 11].



(a) (b)

Fig. 1. (a) ITHC and a rendezvous between climbers; (b) A TSP 2-OPT butterfly

In our implementation, a single hill climber starts from an initial Hamiltonian-cycle, referred to as a configuration, and attempts to improve it using the 2-OPT improvement procedure. Each exchange of two pairs of edges in the 2-OPT routine is referred to as a butterfly. For each configuration, each climber is evaluating all the possible butterflies and chooses the best one. That is, the one that produces the minimum tour amongst all the possible exchanges. The new tour becomes the next configuration for the climber who climbs this way in the steepest descent until reaching a local minimum. Several tour construction algorithms for the TSP have been thoroughly examined in the context of hill climbing [6]. One of these methods, which is giving good results in many practical applications, is the minimum spanning tour method. In this method, a minimum spanning tour is constructed using an algorithm that is similar to the Kruskal's minimum spanning tree construction algorithm [6, 10, 11]. After obtaining the tour the edge connecting the first and last vertices is added to construct a spanning cycle. The algorithm is efficient and presents low complexity. Nevertheless, the last step, adding the last edge which might be a high weighted edge, is responsible to the fact that the algorithm is sub-optimal. Several other tour construction algorithms are listed in the literature but they are considered inferior in complexity and projected solution quality [6, 10, 11].

2.1 Iterative Hill Climbing in the TSP Search Space

The ITHC in the TSP search space procedure is implemented through iterations on two execution phases: tour construction and tour improvement. Most tour construction algorithms used in hill climbing are not suitable for ITHC because they are only concerned with generating one good starting point while ITHC requires many starting points. Another requirement is that the algorithm will generate enough distinct starting points so that the ITHC will not be initialized with the same starting point or exhaust all the starting points before completing all the iterations. The literature includes only one tour construction algorithm to be used for ITHC in TSP. That is, the random restart (RR) algorithm, which randomly generates initial configurations by selecting a random order for the given set of cities during the tour construction phase. The advantage of random restart is that the initial configurations are dispersed throughout the domain. This provides the opportunity to discover new minimums in a small number of iterations. Nevertheless, the random restart procedure might miss the global minimum. Another problem with random restart is that when the number of cities is large it tends to reduce the locality of reference and diminishes the value of caching. In addition to the random restart procedure, we examined a modified minimum spanning tour procedure. The procedure employs a greedy enumeration (GE) of spanning tours according to their weight and converts them into spanning cycles by adding an edge connecting the first and last vertex. The tour improvement phase is identical to the one described under the hill climbing in the TSP search space section.

3 Caching in the ITHC Procedure

Computer systems exploit two empirical principles: "locality of reference" and "small is fast," and implement a memory hierarchy. Generally, under this scheme, the locality of the current CPU reference word is stored in a small and fast cache [15, 16].

Locality of reference is a statistical property of a given data stream that serves as a measure of redundancy [15, 16]. Temporal locality of reference measures the probability of seeing a particular piece of data that had been used in the recent past, in the near future. Spatial locality predicts the probability of using and reusing data elements that are stored in proximity to each other. Related to the concepts of locality of reference are the terms of hit and miss and the hit/miss rate. A hit occurs when a CPU reference-word is located in the cache. A miss occurs when the word is not in the cache and has to be brought from memory. Thus, strong locality of reference is typically equated with lower miss rates since the data that we are about to encounter should already be stored in the cache if it was seen recently. A hit may incur a minor overhead whereas a miss entails a significant penalty that includes evicting a block from the cache and replacing this block with the block that contains the current locality of reference. Hence, depending on the hit (miss) rate, the hit overhead, the miss penalty, and the relative speed of cache memory compared to main memory, a cache design may speed up the system performance significantly. Indeed in many computer systems the cache operates with a hit ratio of more than 90% and significantly enhances the system performance.

There are three main cache organization methods that relate to the way addresses in main memory are mapped to cache address: direct mapping, set associative mapping, and fully associative mapping [15, 16]. Direct mapping does not enable efficient replacement policies and associative memory is quite expensive. Hence, in practice, set associative mapping which enables implementing replacement policies with a reasonable associativity of 4 to 16. For the experiments performed in this research, set associative mapping is used.

Three basic replacement policies and several dynamic / adaptive combinations of the three are generally considered [15, 16]. The first policy is ‘random replacement’. It randomly chooses the block to be evicted. Next a recency-based method, evicts the least recently used block. This method mainly exploits temporal locality. Finally, a frequency-based method evicts the least frequently used block, thereby exploiting spatial locality. We implemented these three methods.

The main trust of our research is that locality of reference exists in many heuristic search spaces and algorithms. It manifests itself in the form of reuse of intermediate data from previous stages of the search or repeating routes that have been examined in previous stages. For example, in genetic algorithm a large portion of the current population of chromosomes have already been evaluated in previous iterations of the algorithm. In ITHC, the locality is due to the fact that several climbers can independently produce an identical configuration. Specifically, in the ITHC / TSP domain, locality is due to the fact that many climbers may attempt to improve the same configuration. Exploiting this locality via caching can decrease the execution time of a heuristic procedure due to one of two reasons: either the original data is located on a storage medium that takes longer to access than the cache, or the original data must be recalculated before it can be used while accessing the cache is quicker than recalculating the data. In our case, the latter is true. By storing data that has been computed, the need to compute the data again is either eliminated completely or the number of times that the data must be computed is reduced. It may not be possible to completely eliminate the need to compute data redundantly, since in practice the cache is finite and storing every piece of data that is desirable to store is, in most cases, not feasible. Additionally, because cache capacity is limited, choices must be made that determine which data is stored and which data is not stored. These choices are determined by a cache replacement as described above.

In a typical implementation of iterative hill climbing, a single climber can perform many iterations of hill climbing, starting from an initial configuration, referred to as seed, and restarting whenever reaching a local optimum. Alternatively, a multitude of climbers can work in parallel trying to find the best local optimum. The addition of a cache to the algorithm changes the restart condition. When a climber generates a new configuration through improvement of an existing configuration she first checks the cache to see if the configuration is stored there. If it is, then the climber restarts. Otherwise, the climber implements a replacement procedure, which results in a decision of whether to store the new configuration in the cache and the specific storage location and eviction.

4 Experiments and Results

Several experiments aimed to assess the locality of reference, the miss ratio, and the overall execution speedup, were performed. Results of experiments with a small number of cities (under 13) where compared to the results obtained through an exhaustive search. In these cases the global optimum was found again and again in many of the ITHC iterations.

Figure 2 depicts the results of temporal and spatial locality in a 20 and 26-city problem. Overall, many scenarios with different number of cities and initialization were assessed through histograms of the reference location (specific configuration) and the time between consecutive references. The figure shows a very high degree of locality. That is, more than 90% of the time/space references are localized close to the global optimum and occur within short time intervals. The cycle initialization method used is the minimum spanning tour (GE). In the case of random restart (RR), the locality diminishes with a larger number of cities (about 18).

Figure 3a shows the trends in miss ratio as a function of the number of sets for the three replacement policies applied to an 8-way set associative cache. The conclusion from that figure is that the 8-way associativity is optimal at around 8192 sets. Other experiments demonstrated that 16-way associativity is optimal at 4096 sets and 4-way associativity is optimal with 16384 sets.

Finally, a set of experiments measuring the speedup obtained by different cache replacement policies, initial configuration generation, and restart methods, as well as different number of cities were performed. The speedup is measured through a highly

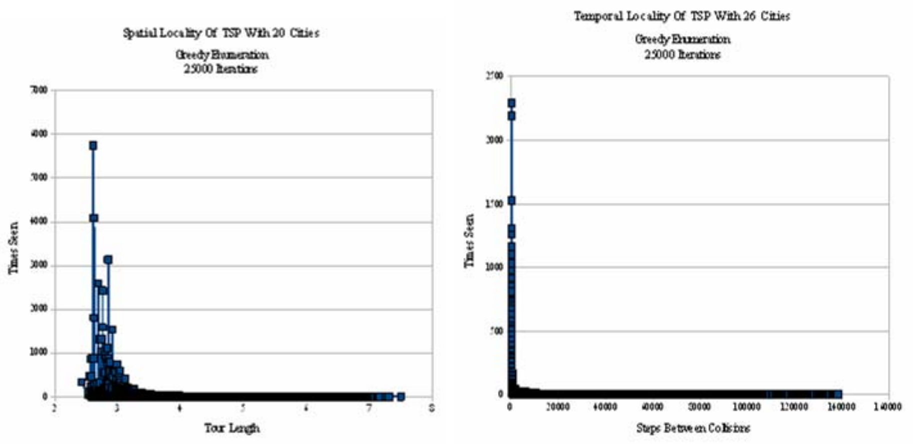
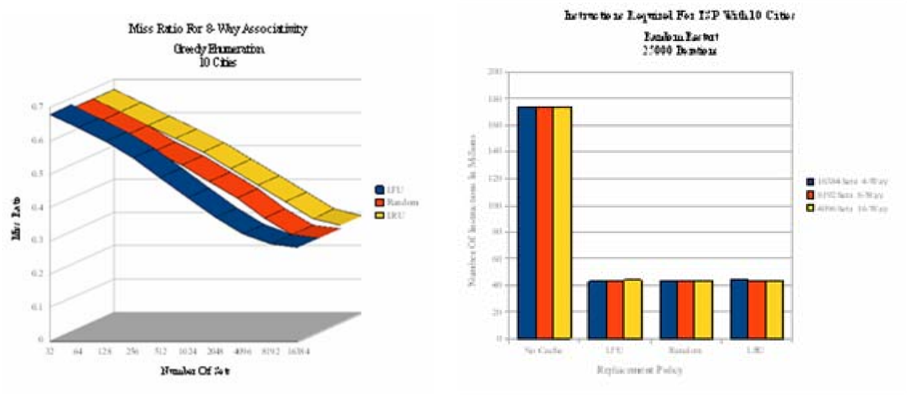
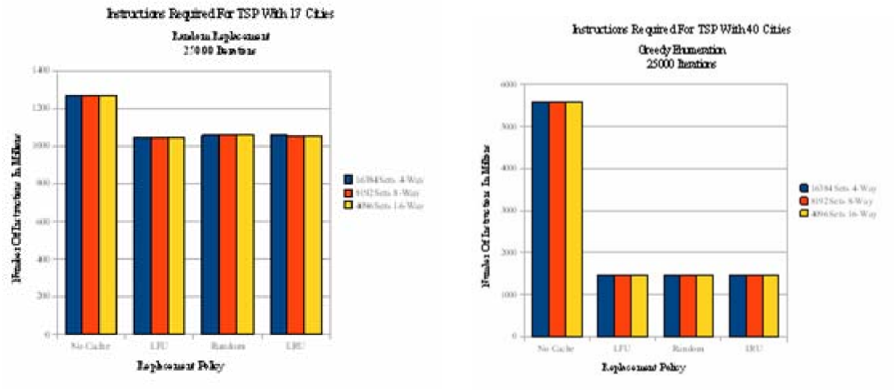


Fig. 2. Spatial locality (*left*) and temporal locality (*right*) in a 20 / 26 city problem



(a) (b)

Fig. 3. (a) (*left*): Miss Ratio of LFU (*blue – left most*), random (*orange - middle*), and LRU (*yellow- right most*), replacement policies with an 8-way set associative cache; (b) (*right*): Number of instructions with and without cache for a 10-city problem and LRU, Random, and LFU replacement policies and random restart



(a) (b)

Fig. 4. (a) (*left*): Number of instructions with and without cache for a 17-city problem (RR). (b) (*right*): Number of instructions for a 40-city problem (GE).

accurate estimation of the number of instructions (hence cycles) required to implement a butterfly. It takes into account hit overheads as well as miss penalty and compares the number of butterflies with and without cache. Figure 3b shows a speedup of more than 3x, obtained through RR initiation with 10 cities.

As Figure 4a shows, the speedup of RR diminishes with a higher number of cities and is only about 20% when RR is applied to a 17-city problem. On the other hand, greedy enumeration is found to be a highly efficient method. It obtains Hamiltonian-cycles that are about 10% shorter than cycles obtained by RR. It also outperforms RR by a factor of 2x (measured in number of butterflies) in execution time. The most important and interesting results of the research are that the cached GE version obtains the same results as the non-cached GE (in terms of cycle length), while showing a speedup of over 4x compared to the non-cached GE. Figure 4b shows a speedup of more than 4x, obtained with GE in a problem with 40 cities. So far, we have addressed problems with up to 80 cities and obtained a speedup of about 4x.

5 Conclusions and Further Research

Overall, this research shows that adding a cache to the ITHC can result in significant speedup. The data presented in this research shows that the greedy enumeration seeding method outperforms the random restart method with and without a cache. On the other hand, there are cases where adding a cache to ITHC results in a decrease in performance compared to a non-cached version. This happens when random initialization is used in a relatively large size problem. For relatively large cache sizes, all of the cache replacement policies perform nearly identically. However, when the size of the cache is small the LFU exhibits the best performance.

The GE is a localized search that concentrates on the search space area that is presumed to be close to the minimum. Nevertheless, if the actual optimum is in a completely different area in the space, the GE is not likely to identify it. The random restart has an advantage that it is not tied to a local area in the search space. Nevertheless, if the search area is large, then the probability of finding good solutions with RR is small. We are currently developing several effective and computationally efficient seeding methods, including a nearest neighbor method and a hub method [11]. We are also experimenting with a combination of random restart and deterministic seeding assuming that this combination can bring the benefits of both methods and enable effective utilization of the cache with relatively large problems (more than 50 cities). Additional items that can be further explored are problems with “huge” number of cities, the usage of Bloom filter as the cache, cache coherency for distributed ITHC, additional seeding methods, additional heuristic search techniques, and different search domains.

References

1. Garey, M.R., Johnson, D.S.: Computers and intractability: A guide to the theory of NP-completeness. W.H. Freeman, New York (1979)
2. Pearl, J.: Heuristics: Intelligent search strategies for computer problem solving. Addison-Wesley, Reading (1984)
3. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220, 671–680 (1983)
4. Vose, M.D.: The simple genetic algorithm: Foundations and theory. MIT Press, Cambridge (1999)
5. Russell, S.J., Norvig, P.: Artificial intelligence: A modern approach, 2nd edn. Prentice-Hall, Englewood Cliffs (1995)
6. Xi, B., Liu, Z., Raghavachari, M., Xia, C., Zhang, L.: A smart hill-climbing algorithm for application server configuration. In: Proceedings of the 13th international conference on world wide web, pp. 287–296 (2004)

7. Hertel, P., Pitassi, T.: An exponential time/space speedup for resolution. Electronic Colloquium on Computational Complexity 46, 1–25 (2007)
8. Allen, D., Darwiche, A.: Optimal time-space tradeoff in probabilistic inference. In: Proceedings of the 21st international joint conference on artificial intelligence, pp. 969–975 (2003)
9. Ganapathy, G., Ramachandran, V., Warnow, T.: Better hill-climbing searches for parsimony. Algorithms in Bioinformatics, 245–258 (2004)
10. Johnson, D.S., McGeoch, L.A.: The traveling salesman problem: A case study in local optimization. Local search in combinatorial optimization, 215–310 (1997)
11. Applegate, D.L., Bixby, R.E., Vasek, C., Cook, W.J.: The traveling salesman problem: A computational study. Princeton University Press, Princeton (2007)
12. Zilberstein, S.: Using anytime algorithms in intelligent systems. AI Magazine 17(3), 73–83 (1996)
13. McMahan, H., Gordon, G.: A fast bundle-based anytime algorithm for poker and other convex games. In: Proceedings of the 11th international conference on artificial intelligence and statistics, pp. 323–330 (2007)
14. Ramos, T., Cozman, G.: Anytime anyspace probabilistic inference. International Journal of Approximate Reasoning 38(1), 53–80 (2005)
15. Stone, H.S.: High-performance computer architecture. Prentice-Hall, Englewood Cliffs (1993)
16. Vanichpun, S., Malkowski, A.M.: Comparing strength of locality of reference – popularity, majorization, and some folk theorems. In: 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 838–839 (2004)

On the Continuous Control of the Acrobot via Computational Intelligence

Sam Chau Duong[1], Hiroshi Kinjo[2], Eiho Uezato[2], and Tetsuhiko Yamamoto[3]

[1] Graduate School of Engineering and Science, University of the Ryukyus
Senbaru 1, Nishihara, Okinawa 903–0213, Japan
[2] Faculty of Engineering, University of the Ryukyus
Senbaru 1, Nishihara, Okinawa 903–0213, Japan
[3] Tokushima Technology College
Itano-gun, Tokushima 779–0108, Japan

Abstract. The focus of this work is the continuous control of the Acrobot under limited-torque condition. By utilizing neural network (NN) and genetic algorithm (GA), a global controller is constructed in order to handle both swing-up and balancing control stages of the Acrobot without the need of different control strategies for the two processes. Based on given control timings, two different evaluation functions are introduced, one being continuous evaluation and the other multi-point based evaluation. In order to improve the system performance, an enhanced GA is proposed which recovers the diversity of population when it tends to be lost by applying an adaptive mutation operator based on a convergence index that reflects the diversity of population in GA. To verify the system performance, numerical simulations are implemented with different timing constraints. Comparisons between the proposed GA with the conventional method as well as between the two evaluation schemes are also provided. Simulation results show that the proposed GA has good performance and the neurocontrol system is able to control the Acrobot effectively by either one of the two evaluation schemes.

Keywords: Acrobot, underactuated system, intelligent control, neurocontroller, genetic algorithms, adaptive mutation.

1 Introduction

For the last several years, the Acrobot has been attracting growing interest as one of challenging robotic control problems. The difficulty in control of the Acrobot arises from the fact that it is a highly nonlinear underactuated system, which processes fewer actuators than degrees of freedom, and exhibits nonholonomic behavior and lack of feedback linearizability [1]. On the other hand, as is well known, nonholonomic systems cannot be stabilized by continuous and time-invariant state feedback control.

The control problem of the Acrobot involves two control processes. The first stage is swing-up control where the goal is to force the Acrobot to reach its vertical upright position with near zero velocity on both links, and the second

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 231–241, 2009.

process is stabilizing or balancing control which is used to maintain the system in the inverted position. While the balancing control is easier as it can be linearized, the swing-up control has been the main focus of much research.

Various control methods have been developed for the Acrobot. Spong M.W. *et al.* [1], [2] have focused on the development of classical control methods for the system. In [1] he presented two different partial feedback linearizing controllers for swing-up control and a linear quadratic regulator (LQR) for stabilizing. Reinforcement learning (RL) approaches [3] -[5] have also been applied effectively for the Acrobot. To overcome the disadvantage of RL techniques in the case of continuous and nonlinear systems, Yoshimoto J. *et al.* [5] proposed an RL scheme combining multiple incomplete controllers for the problem, each of which is constructed based on traditional control theory and performs a specific role. It seems that the first results on the use of fuzzy and genetic algorithms (GAs) for the Acrobot control were reported by Brown S.C. and Passino K.M. [6] who developed several control methods for swing-up and balancing using both classical and fuzzy approaches. Two GAs were then utilized for tuning the designed controllers. Other works on the application of fuzzy approach to this problem can be found in the research devoted by Smith M.H. and his colleagues [7], [8].

Most previous work involved switch controllers that used different control strategies for the Acrobot, i.e. one controller for swing-up and another for balancing. Generally, switch control methods are required to satisfy certain strict criteria to turn to stabilizing control, thus making it difficult to determine the switching time in advance. Also, it is hard to achieve smooth control due to the switch between distinct controllers. In a previous study, we have presented a switch control method using neurocontroller (NC) optimized by GA for this complex problem [9]. Although good control performance was obtained, there is still room for improvement for the system performance as not many NCs could be evolved.

In this paper, our aim is to construct a continuous control strategy that uses only one NC (called global NC) to handle both control stages of the Acrobot with prior timing for each stage. The control problem is also considered under limited torque condition, which presents interesting and important material for research but has not been always studied adequately in literature. In order to improve the system performance, we also propose a GA that has mutation rate changes dynamically based on what we term as a convergence index. Although it is treated only as a secondary operator in the traditional GAs, mutation operator has attracted much attention since it has an important role in GAs for introducing diversity into the population [10]-[12].

The remainder of this article is organized as follows. Sect. 2 introduces the model and dynamics of the Acrobot. In Sect. 3 we present the design of a neurocontrol system optimized by GA with two different evaluation functions. The proposed GA is introduced in Sect. 4. Sect. 5 contains numerical simulations implemented with different timings for comparing between the proposed GA and the canonical GA as well as between two evaluation schemes. Lastly, we discuss and conclude this research in Sect. 6.

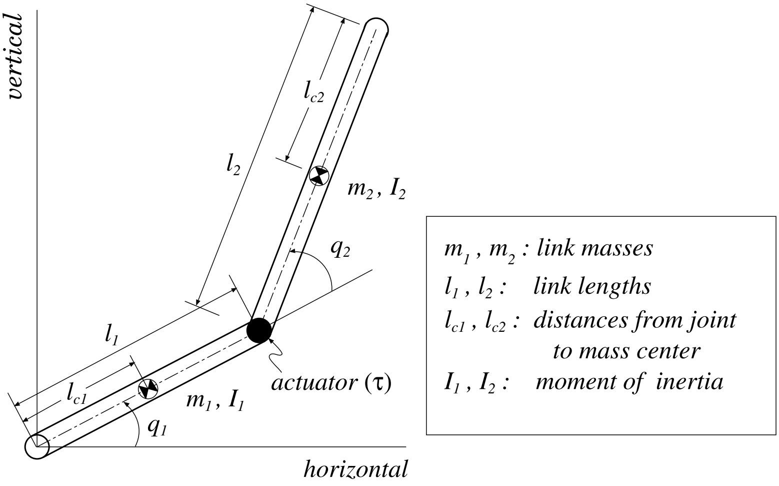


Fig. 1. The Acrobot system

2 The Acrobot System and Dynamics

The Acrobot, which is roughly analogous to a human gymnast swinging on a high bar, is an underactuated system having two links (each with one degree-of-freedom) and a single actuator at the second joint but no actuator at the first joint. The model of the Acrobot and its basic physical parameters are illustrated in Fig.1, where the joint angles q_1 and q_2 serve as generalized system coordinates and the torque τ produced by the actuator is the input of the system. The dynamic equations of the system are as follows:

$$d_{11}\ddot{q}_1 + d_{12}\ddot{q}_2 + h_1 + \phi_1 = 0 \tag{1}$$

$$d_{21}\ddot{q}_1 + d_{22}\ddot{q}_2 + h_2 + \phi_2 = \tau \tag{2}$$

where the coefficients in Eq. 1 and 2 are defined as:

$$\left.\begin{aligned}
d_{11} &= m_1 l_{c1}^2 + m_2 \left\{ l_1^2 + l_{c1}^2 + 2 l_1 l_{c2} \cos(q_2) \right\} + I_1 + I_2 \\
d_{12} &= m_2 \left\{ l_{c2}^2 + l_1 l_{c2} \cos(q_2) \right\} + I_2 \\
d_{21} &= d_{12} \\
d_{22} &= m_2 l_{c2}^2 + I_2 \\
h_1 &= -m_2 l_1 l_{c2} \sin(q_2)\, \dot{q}_2^2 - 2 m_2 l_1 l_{c2} \sin(q_2)\, \dot{q}_2 \dot{q}_1 \\
\phi_1 &= (m_1 l_{c1} + m_2 l_1)\, g \cos(q_1) + m_2 l_{c2} g \cos(q_1 + q_2) \\
h_2 &= m_2 l_1 l_{c2} \sin(q_2)\, \dot{q}_1^2 \\
\phi_2 &= m_2 l_{c2} g \cos(q_1 + q_2)
\end{aligned}\right\} \tag{3}$$

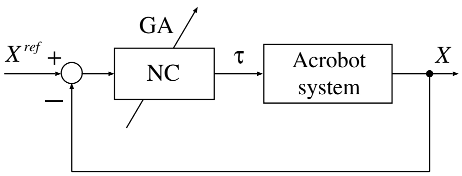


Fig. 2. Control system

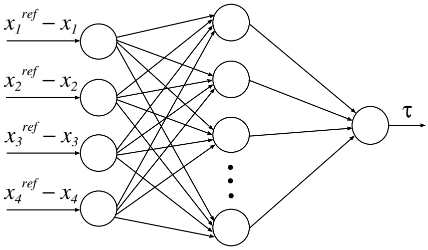


Fig. 3. Neurocontroller

3 Control System Design

3.1 Control System

Recently, the applications of computational intelligence techniques have been applied extensively in the field of control engineering, especially for the systems that cannot be easily controlled by conventional means. This subsection presents the design of a control system using NC evolved by GA for the Acrobot.

Let $x_1 = q_1$, $x_2 = q_2$, $x_3 = \dot{q}_1$, $x_4 = \dot{q}_2$, the state of the system is defined as $X = [x_1, x_2, x_3, x_4]^T$. The task of the controller is to swing the Acrobot from its stable downward position, which has initial state $X^{init} = [-\pi/2,\ 0, 0, 0]^T$, to its unstable inverted position and balance it about the vertical that has state variable $X^{ref} = \left[x_1^{ref}, x_2^{ref}, x_3^{ref}, x_4^{ref}\right]^T = [\pi/2,\ 0, 0, 0]^T$.

Figure 2 shows the control system, a state feedback controller, designed for the Acrobot. Fig. 3 illustrates the structure of the NC which is a three-layer feedforward NN consisting of input layer, hidden layer, and output layer.

3.2 Genetic Algorithm in the Controller Design

In this study, GA is utilized to search for the optimal sets of the connection weights of the NC, each of the weights is transformed into a genetic code encoded by 16−bit binary code.

The algorithm flow of GA is as follows:

- **Step 1:** Initializing a population of NCs with sets of connection weights drawn randomly from a uniform distribution.
- **Step 2:** Control simulations are performed using the NCs. The control performance of each NC is then evaluated.
- **Step 3:** Offspring NCs are produced by the parent NCs which are selected based on the evaluated performances.
- **Step 4:** Control simulations are implemented for the offspring NCs, and their performances are then evaluated.
- **Step 5:** Ranking the NC individuals in the pool of all parent NCs and offspring NCs. The poor-fitness individuals are eliminated from the population.

– **Step 6:** Stop if the termination condition is satisfied. Otherwise go back to Step 3.

In GA, Roulette wheel technique is used to select parents for reproduction in proportion to their relative fitness, which is defined as:

$$F^{(p)} = \frac{1}{1 + E^{(p)}} , \qquad p = 1, 2, \cdots N \tag{4}$$

where $E^{(p)}$ is the error (or evaluation) function value of the p^{th} NC individual, and N is the population size.

3.3 Evaluation Functions for the Continuous Control of the Acrobot

Suppose that the times for swing-up and stabilizing are defined respectively in advance as t_s and t_b. In our previous study [9], we used an evaluation function based on the final state alone (at the time t_s) for the NC to perform only swing-up action. In this paper, in order to maintain the obtained state after swing-up, we apply a continuous evaluation function as follows:

$$E^{(p)}_{\text{cont.}} = \int_{t_s}^{t_s+t_b} \left\{ \sum_{i=1}^{4} Q_i \left(x_i^{ref} - x_i(t) \right)^2 \right\} dt \tag{5}$$

where Q_i are the weight coefficients, $x_i(t)$ are the state variables at the time t. In the total control simulation time $(t_s + t_b)$, t_s is the time spent for swing-up where the NC is expected to acquire swing-up ability and eventually to force the Acrobot to reach the desired configuration at final state, i.e., the state at the time t_s. This state is then continued in t_b seconds for balancing by applying the evaluation in GA to this entire period of time.

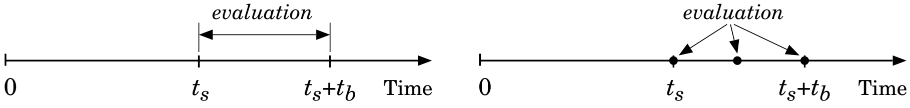
In our opinion, the use of the continuous evaluation in Eq. 5 would result in the difficulty in evolving the NC by GA and may require a large computational cost. Therefore, we also propose a multi-point (or discrete) evaluation function as:

$$E^{(p)}_{\text{disc.}} = \sum_{k} \left\{ \sum_{i=1}^{4} Q_i \left(x_i^{ref} - x_i(k) \right)^2 \right\} \tag{6}$$

where $x_i(k)$ are the state variables of the system at the time point k taken into the evaluation. For example, in this study we shall use three points $k = \{t_s,\ (t_s + t_b/2),\ (t_s + t_b)\}$ for evaluation in the hope that the evaluated points would result in straight lines for balancing in the control simulation curves. The plot of the proposed evaluation schemes is shown in Fig. 4.

4 An Improved Genetic Algorithm with an Adaptive Mutation Operator

Traditionally, the mutation rate in a GA is kept constant during the evolution process. In this article, we propose an improved GA using an adaptive mutation



(a) Continuous evaluation (b) Multi-point based evaluation

Fig. 4. Evaluation schemes for the NC

operator. That is, the mutation rate changes according to a convergence index $\xi = \frac{E_{\text{best}}}{E_{\text{avr}}}$, where E_{best} is the error value of the best NC in the population and E_{avr} is the average error value of all NC individuals in a generation. We consider that convergence occurs when this index is larger than a certain value (in other words, most NCs tend to be similar to the best NC or the population diversity is apt to be lost).

Based on the defined convergence index ξ, the rate of mutation r_{mut} is altered dynamically according to the following rule:

$$r_{\text{mut}} = \begin{cases} 0.5, & \text{if} \quad \xi > 0.6, \\ 0.1, & \text{otherwise} \end{cases} \tag{7}$$

By such a rule, a high rate of mutation is applied when the diversity of the NC population tends to be lost.

5 Numerical Simulations

5.1 Test Design and Parameters

In this study, the fourth–order Runge–Kutta technique is used with step size $h = 0.005$ seconds. The parameters of the Acrobot are depicted in Table 1 (refer to [1]), and the parameters of GAs are shown in Table 2. The mutation rate in the conventional GA is kept constant at 0.3. The linear activation $f(x) = x$ is used in input and output layers of the NC. For hidden layer, the hyperbolic tangent activation function $f(x) = \tanh(x)$ is applied and 6 neurons are used (namely, a 4–6–1 NN structure is utilized). The initial NC connection weights are drawn randomly from a uniform distribution from the range [−1.0, 1.0]. While it is desirable to tune for the most suitable test parameters for the purpose of enhancing system performance, we shall not carry out such large-scale work due to the constraints of this article. Rather, the selected parameters are those that provide the best performance among the values we have tested (empirically).

Table 1. Parameters of the Acrobot system

Parameter	Link 1	Link 2
Mass [kg]	m_1=1	m_2=1
Length [m]	l_1=1	l_2=2
Center of mass [m]	l_{c1}=0.5	l_{c2}=1
Inertial moment [kgm^2]	I_1=0.083	I_2=0.33

Table 2. Parameters of GAs

Parameter	Value/scheme
Population N	200
No. of offspring	$0.6 \times N$
No. of generations	3000
Bit number	16
Solution range of connection weights	$[-20, 20]$
Crossover operator	Two-point
Mutation operator	One-bit
Selection scheme	Roulette wheel
Q_i $(i = 1, 2, 3, 4)$	1.0

Table 3. Rates of successfully-evolved NCs [%]

(a) Using the conventional GA

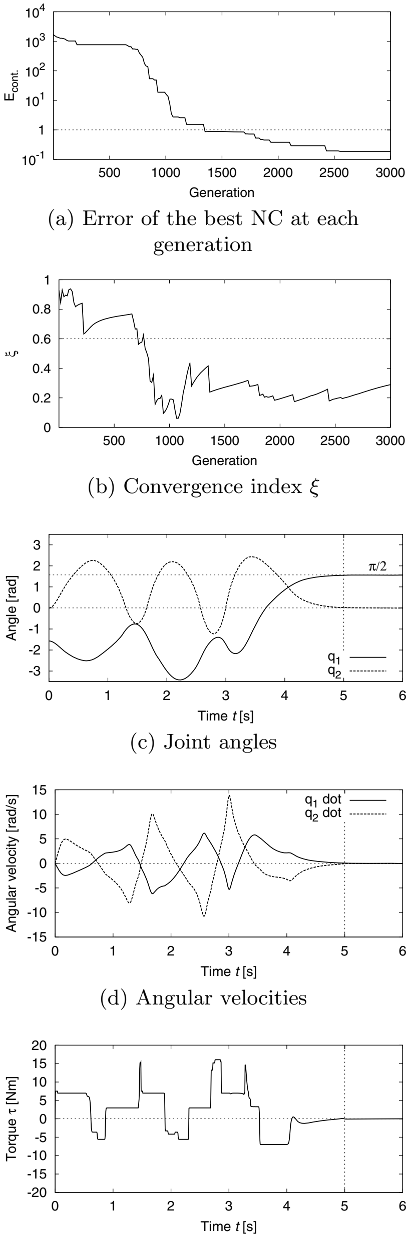
Swing-up time t_s [second]		1.0	2.0	3.0	4.0	5.0
Continuously-evaluated NC	$t_b = 1$ s	0	4	8	8	10
	t_b =2 s	0	2	4	6	6
Discretely-evaluated NC	$t_b = 1$ s	0	8	14	18	22
	t_b =2 s	0	6	8	12	14

(b) Using the proposed GA

Swing-up time t_s [second]		1.0	2.0	3.0	4.0	5.0
Continuously-evaluated NC	$t_b = 1$ s	0	8	12	14	20
	t_b =2 s	0	4	6	10	12
Discretely-evaluated NC	$t_b = 1$ s	0	14	24	32	40
	t_b =2 s	0	10	12	18	22

We also consider the Acrobot control problem under limited torque condition by using penalty factor. That is, if the torque τ output by an NC violates the condition $|\tau| > 20$Nm, a penalty value $E_{\text{pnt}} = 10^6$ will be added to the evaluation function of that NC.

For comparisons between the proposed GA and the canonical GA as well as between the two evaluation schemes, various simulations will be accomplished using distinct prespecified timings. Since the performance of a GA is highly dependent on the initial random weights, 50 test iterations will be implemented using different initial populations for each timing condition. After running GA, if an NC can obtain an error value less than a required accuracy E_{suc}, the NC is considered to be successfully evolved. The system performance is evaluated by the rate of successfully-evolved NCs calculated over the 50 replications. Regarding to the evaluation term, we set $E_{\text{suc}} = \frac{t_b}{h} \times E_{\text{1point}}$ for the continuous evaluation case (Eq. 5), and $E_{\text{suc}} = n_k \times E_{\text{1point}}$ for the multi-point based evaluation case (Eq. 6), where $E_{\text{1point}} = 0.005$ is the required accuracy for one single point and n_k is number of points k taken into evaluation in Eq. 6 (here, $n_k = 3$).



(a) Error of the best NC at each generation

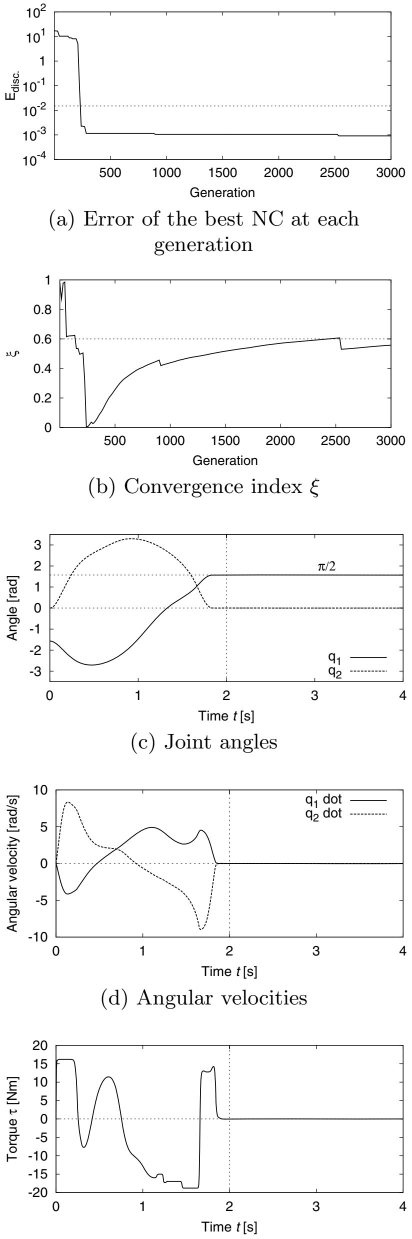
(b) Convergence index ξ

(c) Joint angles

(d) Angular velocities

(e) Input torque τ

Fig. 5. Control results of a continuously-evaluated NC with $t_s = 5\text{s}$, $t_b = 1\text{s}$



(a) Error of the best NC at each generation

(b) Convergence index ξ

(c) Joint angles

(d) Angular velocities

(e) Input torque τ

Fig. 6. Control results of a discretely-evaluated NC with $t_s = 2\text{s}$, $t_b = 2\text{s}$

5.2 Simulation Results

Table 3 shows the results on the rates of successfully-evolved NCs while comparing the proposed GA to the canonical GA using the two evaluation schemes and different timing conditions. We can observe that, the proposed GA has better performance than the conventional GA. In addition, the use of the discrete-point evaluation function results in the ease in evolving the NC rather than using the continuous function. Also, the difficulty in evolving the NC increases with decreasing the swing-up time t_s and/or increasing the evaluation term t_b.

For instance, control simulations are performed using the resulting NCs evolved by the proposed GA with $t_s = 5$s, $t_b = 1$s (for the case of the continuously-evaluated NC), and $t_s = 2$s, $t_b = 2$s (for the case of the discretely-evaluated NC). The control results are shown in Figs. 5 and 6. It is clear that both control stages are handled well by the NCs. In the first test of the continuously-evaluated NC with $t_s = 5$s and $t_b = 1$s, swing-up motion was iterated several times to gradually gain energy by oscillating the second link in synchronization with the first one. As the result, the NC could swing up the Acrobot with small torque (see Fig. 5). When using the discretely-evaluated NC with harder timing constraints ($t_s = 2$s and $t_b = 2$s), as shown in Fig. 6, one swing-up motion was performed with slightly larger torque.

6 Discussion and Conclusion

Following the trend of current research employing computational intelligence approaches for control systems design, we have presented a continuous control method with applications of NN and GAs for the Acrobot in this paper.

An improved GA has been introduced that can recover the diversity of population when it tends to be lost by applying an adaptive mutation operator. To handle both swing-up and balancing controls by one global controller, two different evaluation functions have been introduced. As seen in Table 3, the proposed GA is able to provide better performance than conventional means. Also, the use of the continuous evaluation function causes difficulty in evolving the NC since the control system is required to maintain small error in every time step and it results in large values for the accumulated error function. Certainly, the point-to-point evaluation as Eq. 6 does not always guarantee a smooth balancing control, and the number of evaluation point n_k might affect the system performance. That is, many evaluation points may provide better balancing control but may lead to a high computational cost and difficulty in evolving the NC by GA.

Generally, switch-condition based approaches cannot provide prior information of control time, particularly the swing-up time. Also, it is hard to obtain smooth control due to the switch between distinct controllers [9] as the balancing controller usually needs some time to stabilize the system properly. In this paper we did not use any swing-up or switching conditions for the two stages. Rather, the control system was tested with several timings defined in advance and the NC is expected to perform swing-up and balancing within the given conditions.

Using the NCs evolved with two different timing constraints, the control simulations have also been demonstrated. It can be observed from the results that the NCs could control the Acrobot effectively within the limited torque. In both cases, the desired stable state could be obtained almost right after swing-up phase.

According to the study, it appeared to be hard to evolve the NC. This is either because of the difficulty of the problem and/or the limited torque condition and long evaluation term. In addition, the NC accomplishes swing-up during the time t_s without any rule-based guide or knowledge of how to perform swinging motion/trajectory. The NC is expected to complete the goal based only on the final desired configuration described in the evaluation functions. Also, we consider that the wide motion range of the two links of the Acrobot results in several local optima that GAs are easily trapped into. Thus, a rich population diversity might be needed for GA to explore the search space. However, applying a too high mutation rate to the entire evolution process may lead to the loss of good solutions (as many good genes are altered). This causes premature convergence. In our proposed GA, high rate of mutation is applied only when ξ is large ($\xi > 0.6$ or the population diversity tends to be lost), the diversity is then recovered and GA continues searching for a better NC. As a result, the value of the convergence index ξ decreases. A lower mutation rate ($r_{\text{mut}} = 0.1$) is then applied to maintain good solutions but it is still large enough to keep the effect of enriching the population diversity (see Figs. 5 (b) and 6 (b) for examples).

Although good results have been achieved by the proposed methods, it is necessary to carry out a real implementation for the system in future work. In such a case, the emergence of noise and nonlinearities such as mechanical friction is a challenging problem. Thus, it may be necessary to train the resulting NC again in the real environment. In addition, a good generalization ability of the NC may be useful. In this study GAs have been employed for evolving the NC with a single initial state (the ideal vertical downward position). It would be positive if the NC is evolved with several initial states which are slightly variant from the ideal one.

References

1. Spong, M.W.: The Swing Up Control Problem for the Acrobot. IEEE Control Systems Magazine 15(1), 49–55 (1995)
2. Spong, M.W.: Underactuated Mechanical Systems. In: Siciliano, B., Valavanis, K.P. (eds.) Control Problems in Robotics and Automation. Lecture Notes in Control and Information Sciences, vol. 230. Springer, London (1997)
3. Sutton, R.S.: Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding. In: Advances in Neural Information Processing Systems 8, pp. 1038–1044. MIT Press, Cambridge (1996)
4. Boone, G.: Efficient Reinforcement Learning: Model-Based Acrobot Control. In: IEEE International Conference on Robotics and Automation, vol. 1, pp. 229–234 (1997)
5. Yoshimoto, J., Nishimura, M., Tokida, Y., Ishii, S.: Acrobot Control by Learning the Switching of Multiple Controllers. Artificial Life and Robotics 9, 67–71 (2005)

6. Brown, S.C., Passino, K.M.: Intelligent Control for an Acrobot. Journal of Intelligent Systems 18, 209–248 (1997)
7. Smith, M.H., Lee, M.A., Ulieru, M., Gruver, W.A.: Design Limitations of PD Versus Fuzzy Controllers for the Acrobot. In: IEEE International Conference on Robotics and Automation, vol. 2, pp. 1130–1135 (1997)
8. Smith, M.H., Lee, M.A., Gruver, W.A.: Dynamic Fuzzy Control and System Stability for the Acrobot. In: Fuzzy Systems Proceedings, IEEE World Congress on Computational Intelligence, vol. 1, pp. 286–291 (1998)
9. Duong, S.C., Kinjo, H., Uezato, E., Yamamoto, T.: A Switch Controller Design for the Acrobot using Neural Network and Genetic Algorithm. In: Proceedings of the 10th IEEE International Conference on Control, Automation, Robotics and Vision, ICARCV 2008, Hanoi, Vietnam, pp. 1540–1544 (2008)
10. Duong, S.C., Kinjo, H., Uezato, E., Yamamoto, T.: Applications of Sinusoidal Neural Network and Momentum Genetic Algorithm to Two-wheel Vehicle Regulating Problem. IEEJ Trans. Electrical and Electronic Engineering 3(1), 92–99 (2007)
11. Srinivas, M., Patnaik, L.M.: Genetic Algorithms: A Survey. IEEE Computer 27(6), 17–26 (1994)
12. De Falco, I., Della Cioppa, A., Tarantino, E.: Mutation-based Genetic Algorithm: Performance Evaluation. Applied Soft Computing 1, 285–299 (2002)

Evaluating and Enhancing the Long-Term Competitiveness of a Semiconductor Product

Yu-Cheng Lin[1], Toly Chen[2,*], and Kun-Tai Li[2]

[1] Department of Industrial Engineering and Management,
Overseas Chinese Institute of Technology, 100, Chiaokwang Road,
Taichung City, Taiwan
yclin@ocit.edu.tw
[2] Department of Industrial Engineering and Systems Management,
Feng Chia University, 100, Wenhwa Road, Seatwen, Taichung City, Taiwan
tolychen@ms37.hinet.net

Abstract. Yield is undoubtedly the most critical factor to the competitiveness of a product in a semiconductor manufacturing factory. Therefore, evaluating the competitiveness of a product with its yield is a reasonable idea. For this purpose, Chen's approach is extended in this study to evaluate the long-term competitiveness of a product through yield learning modeling in various ways. Subsequently, to enhance the long-term competitiveness of the product, capacity re-allocation is shown to be helpful. The effects are modeled. Finally, a fuzzy nonlinear programming (FNP) model is constructed to optimize the performance. A practical example is used to demonstrate the proposed methodology.

1 Introduction

To survive in the highly competitive semiconductor industry, every semiconductor manufacturer enhances its own competence in several ways. For example, Armstrong [1] proposed the four principles for competitive semiconductor manufacturing as continuous measurable improvement, statistical thinking, constraint-focus, and people development/empowerment. Jenkins et al. [13] emphasized the importance of quality, and introduced some ways of designing quality into the product and the process. In Peng and Chien [20], the focus was on how to create values for customers. Reducing cycle time, producing high quality products, on-time delivery, continual reduction of costs and improving efficiency were considered as the most direct and effective ways. Recently, Liao and Hu [15] indicated that knowledge transfer is a vital factor to the competitiveness of a semiconductor manufacturer. As a summary, product quality/yield is always considered as one of the most basic and important factors to the competitiveness of a semiconductor manufacturing factory [3, 6, 9]. A high yield means that a high output quantity can be produced with the same input quantity (and total costs), which results in a low unit cost. Besides, in capacity planning, the majority of capacity should be devoted to products with high yields and/or prices. Incorrectly releasing raw wafers to produce low-yield products will inevitably increase the

[*] Corresponding author.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 242–251, 2009.

average unit cost [6]. Further, an acceptable product yield is a prerequisite for volume production [11].

For these purposes, an accurate prediction of yield is necessary. There are two viewpoints for that – the macro yield modeling (MaYM) viewpoint and the micro yield modeling (MiYM) viewpoint [6]:

1. Micro yield modeling (MiYM): Methods using critical device area, parametric sensitivity, redundancy effect, and other factors which rely on a detailed understanding of circuit design to estimate the effects of different classes of defects, process variability, and layout variation on yield. In this category, the composite yield model and the layered yield model were frequently applied [12]. However, semiconductor yield forecasting is still a very difficult task from the MiYM viewpoint. For instance, Hsieh and Tong [12] mentioned that even through the defect count, the defect size, and the defect distribution were the same, the yield loss of the complicate manufactured product was less than that of the simple manufactured product. Besides, defect clustering increases with increasing wafer size. As a result, the conventional Poisson yield model used in this category tends to underestimate yield [23].
2. Macro yield modeling (MaYM): Methods using die size, device density, and other large-scale a priori factors to predict yields for new designs. One common technique in MaYM is learning and transition analysis (LTA), in which a learning model is used to predict the future yield of a product, and the transition of learning effects is also investigated. From the MaYM viewpoint, yield prediction is also a difficult task because yield improvement is a learning process, and the uncertainty and variation inherent in the learning process are not easy to consider. Studies incorporating uncertainty and/or stochasticity into learning phenomenon modeling include Spence [22], Majd and Pindyck [18], Mazzola and McCardle [19], and so on. In the study of Watada, Tanaka, and Shimomura [24], the fuzzy set theory was applied to model learning uncertainty through fitting a possibility regression model. Following Watada et al's study, Chen and Wang [6] proposed a fuzzy yield learning model. Unlike the use of symmetric triangular fuzzy numbers (TFNs) in Watada et al.'s study, the parameters in Chen and Wang's model can have asymmetric shapes.

On the other hand, the involvement of human factors in the learning process often complicates the situation [24], and expert opinions are valuable and effective in handling this situation. Chen and Wang [6] therefore designed two correction functions to incorporate these expert opinions – one for incorporating expert opinions about the asymptotic yield (the final yield) and the other for incorporating expert opinions about the learning constant (the learning speed). On the other hand, if the yield of a product could not reach a certain level (Y_m^*) before a given deadline (m), then the competitiveness of the product will disappear and capacity will be re-allocated to other products. In this respect, a systematic procedure was proposed in Chen [3] to evaluate the mid-term competitiveness of a semiconductor product through yield learning modeling. To prevent that from happening, some managerial actions, e.g. executing quality engineering (QE) projects, quickening the speed of mass production, etc., can be taken to accelerate yield learning [2-3, 11]. After these managerial actions, the yield learning model has to be modified. For this purpose, in Chen [2], Chen and Wang's fuzzy yield learning model was modified to incorporate the effects of such managerial actions.

Existing studies have the following problems that will be tackled in this study:

1. At first, to apply Chen's approach to evaluate the long-term competitiveness of a product, a longer time period than that adopted in Chen's study involving multiple check points has to be observed. If the product is competitive at all check points, then undoubtedly the product has long-term competitiveness. Otherwise, the evaluation results at all check points need to be aggregated.
2. Besides, there is space for improving the long-term competitiveness by, say, capacity re-allocation that can be done at any check point. The effect of capacity re-allocation has to be modeled.
3. In addition, how to establish an efficient capacity re-allocation plan is also a critical task.
4. Further, in Chen's study the competitiveness of a product is expressed with a crisp value, considering the subjective nature of the competitiveness concept using linguistic terms is more appropriate.

2 Evaluating the Mid-Term Competitiveness of a Product

The mid-term competitiveness of a product can be evaluated through yield learning modeling. Considering the uncertainty in yield learning, Chen and Wang's fuzzy yield learning model [6] is adopted as the yield learning model in this study:

$$\tilde{Y}_t = \tilde{Y}_0 e^{-\frac{\tilde{b}}{t}+r(t)} = e^{(y_1-\frac{b_3}{t},\ y_2-\frac{b_2}{t},\ y_3-\frac{b_1}{t})+r(t)}, \tag{1}$$

$\tilde{Y}_t$: the yield at time period t;

$\tilde{Y}_0$: the asymptotic/final yield, which is a real-valued function of point defect density per unit area, chip area, and a set of parameters unique to the specific yield model;

$\tilde{b}$: the learning constant;

$r(t)$: homoscedastical, serially non-correlated error term.

The competitive region is specified as

$$Y_t \geq Y_m^* \text{ when } t \geq m. \tag{2}$$

The mid-term competitiveness, indicated with $\tilde{c} = (c_1, c_2, c_3)$, is evaluated with the area of the competitive region that is under the yield learning curve from time m to $m + \Delta t$ [3]:

$$\begin{aligned}\tilde{c} &= \int_m^{m+\Delta t} \tilde{Y}_t dt - \int_m^{m+\Delta t} Y_m^* dt = \int_m^{m+\Delta t} \tilde{Y}_0 e^{-\frac{\tilde{b}}{t}} dt - Y_m^* \Delta t \\ &= (\int_m^{m+\Delta t} e^{y_1-\frac{b_3}{t}} dt - Y_m^* \Delta t, \int_m^{m+\Delta t} e^{y_2-\frac{b_2}{t}} dt - Y_m^* \Delta t, \int_m^{m+\Delta t} e^{y_3-\frac{b_1}{t}} dt - Y_m^* \Delta t),\end{aligned} \tag{3}$$

where Δt is a small value (≤ 1) indicating the time allowance for achieving the mid-term yield target. If $\Delta t = 1$, then the calculation result ranges from $-Y_m^*$ to $1-Y_m^*$, and is interpreted with some linguistic terms like "Very Competitive (VC)", "Somewhat Competitive (SC)", "Moderate (M)", "Somewhat Uncompetitive (SU)", and "Totally Uncompetitive (TU)". These linguistic terms can be converted into pre-specified fuzzy numbers (see Fig. 1).

The practical example in Table 1 is analyzed. The fitted Chen and Wang's fuzzy yield learning model (satisfied at the level of satisfaction (α) = 0.5) is

$$\tilde{Y}_t = e^{(-0.66-\frac{0.52}{t},\ -0.27-\frac{0.52}{t},\ 0-\frac{0.52}{t})+r(t)}.$$

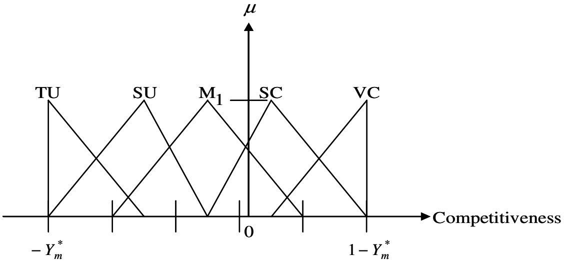


Fig. 1. Linguistic terms for expressing the competitiveness

Table 1. An example

t	1	2	3	4	5	6	7	8	9	10
Y_t	0.373	0.585	0.541	0.741	0.617	0.8	0.712	0.717	0.59	0.724

The calculation procedure refers to Chen and Wang [6]. Assume the competitive region is specified as

$$Y_t \geq 76\% \text{ when } t \geq 15;\ \Delta t = 1.$$

So $m = 15$ and $Y_m^* = 76\%$. Then the mid-term competitiveness of the product can be evaluated as

$$\tilde{c} = (\int_{15}^{16} e^{-0.66-\frac{0.52}{t}} dt - 0.76 \cdot 1, \int_{15}^{16} e^{-0.27-\frac{0.52}{t}} dt - 0.76 \cdot 1, \int_{15}^{16} e^{0-\frac{0.52}{t}} dt - 0.76 \cdot 1)$$
$$= (-0.26,\ -0.02,\ 0.21).$$

The evaluated competitiveness can be most suitably expressed with the linguistic term "SC".

3 Evaluating the Long-Term Competitiveness of a Product

To apply Chen' approach to evaluate the long-term competitiveness of a product, a longer time period than that adopted in Chen's study involving multiple check points has to be observed. There are K check points $m_{(1)}$ ~ $m_{(K)}$. The competitive regions specified at these check points are indicated with $\text{CR}_{(1)}$ ~ $\text{CR}_{(K)}$, respectively:

$$\text{CR}_{(k)}\text{: } Y_t \geq Y^*_{m(k)} \text{ when } t \geq m_{(k)},\ k = 1\text{~}K. \tag{4}$$

The mid-term competitiveness of the product at each check point can be assessed with the procedure introduced previously:

$$\tilde{c}_{(k)} = \int_{m_{(k)}}^{m_{(k)}+\Delta t} \tilde{Y}_t dt - \int_{m_{(k)}}^{m_{(k)}+\Delta t} Y^*_{m(k)} dt,\ k = 1\text{~}K. \tag{5}$$

If all $\tilde{c}_{(k)}$'s are positive, then undoubtedly the product has long-term competitiveness. Otherwise, whether the product will be long-term competitive or not is unclear. To deal with this problem, several ways are proposed to evaluate the long-term competitiveness with $\tilde{c}_{(k)}$'s:

1. The minimal mid-term competitiveness [4]: $\min\limits_{k} \tilde{c}_{(k)}$.
2. The average mid-term competitiveness [4]: $\bar{\tilde{c}}_{(k)} = \sum\limits_{k=1}^{K} \tilde{c}_{(k)} / K$.
3. The weighted average competitiveness [4]: $\sum\limits_{k=1}^{K} \tilde{w}_{(k)} (\times) \tilde{c}_{(k)} (/) \sum\limits_{k=1}^{K} \tilde{w}_{(k)}$, where $\tilde{w}_{(k)}$ is the linguistic assessment for the importance of the competitiveness at check point k, and can be converted into pre-specified fuzzy numbers (see Fig. 2). (×) and (/) denote fuzzy multiplication and division, respectively. The competitiveness of a semiconductor product is usually of unequal importance at different stages in its life cycle [11]. The weighted average competitiveness reflects this consideration.
4. The trend in the mid-term competitiveness: The three measures above are only the statistics about the past. However, the long-term competitiveness considers the performance in the future. For this purpose, the trend in the mid-term competitiveness can be measured, for example, with the fuzzy correlation coefficient (FCC) between the mid-term competitiveness and time:

$$\tilde{R} = \frac{\tilde{S}_{xy}}{\sqrt{\tilde{S}_{xx}\tilde{S}_{yy}}} = \frac{\sum\limits_{\text{all } k} m_{(k)} \tilde{c}_{(k)} (-) K\bar{m}\bar{\tilde{c}}}{\sqrt{\sum\limits_{\text{all } k} m^2_{(k)} - K\bar{m}^2} \sqrt{\sum\limits_{\text{all } k} \tilde{c}^2_{(k)} (-) K\bar{\tilde{c}}^2}}. \tag{6}$$

To derive $\tilde{R}$, the arithmetic for triangular fuzzy numbers can not be applied, and therefore the α-cut operations have to be used:

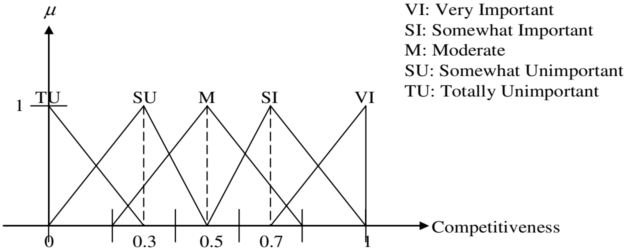


Fig. 2. Linguistic terms for importance

The previous example is again used for demonstration. Assume there are three check points for which the following competitive regions are specified: $\text{CR}_{(1)}$: $Y_t \geq$ 71.5% when $t \geq 13$; $\text{CR}_{(2)}$: $Y_t \geq$ 75% when $t \geq 16$; $\text{CR}_{(3)}$: $Y_t \geq$ 87% when $t \geq 19$. The mid-term competitiveness at each check point is assessed. The results are summarized as follows:

$$\tilde{c}_{(1)} = (-0.22, 0.02, 0.25);\ \tilde{c}_{(2)} = (-0.25, -0.01, 0.22);\ \tilde{c}_{(3)} = (-0.37, -0.13, 0.10).$$

Based on these results, the long-term competitiveness of the product is evaluated in various ways:

1. The minimal mid-term competitiveness = (-0.37, -0.13, 0.10).
2. The average mid-term competitiveness = (-0.28, -0.04, 0.19).
3. The weighted average competitiveness: Assume the weights for the three check points are specified as “Moderate”, “Somewhat Important”, and “Very Important”, respectively. Then the weighted average competitiveness is calculated as (-0.32, -0.06, 0.19) by applying the α-cut operations. The results of the first three measures are compared in Fig. 3, which reveals that the long-term competitiveness of the product is approximately equal to “SC”.
4. The trend in the mid-term competitiveness measured with FCC is shown in Fig. 4, which implies the long-term competitiveness is more uncertain if the entire product life cycle is considered but can be best represented with “totally uncompetitive” or “somewhat uncompetitive”.

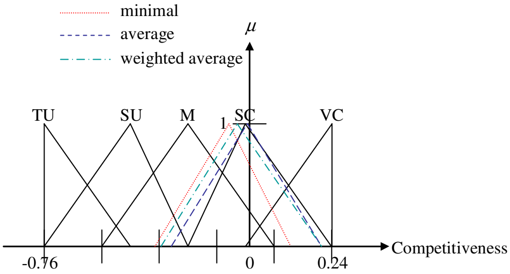


Fig. 3. The evaluated long-term competitiveness

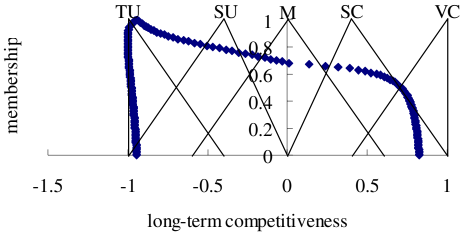


Fig. 4. The evaluation result

4 Re-allocating Capacity to Enhance the Long-Term Competitiveness

The learning speed of a product is proportional to the capacity allocated to the product. The relationship between the evaluated mid-term competitiveness of a product and its learning constant is an exponential function. A large learning constant represents a slow learning speed which results in a low competitiveness. Assume the capacity originally allocated for making the product is to release q wafers per period. After re-allocation the capacity grows to q'. Then the mid-term competitiveness of the product can be elevated to

$$\tilde{c}'_{(k)} = \tilde{Y}_0 \int_{m_{(k)}}^{m_{(k)}+\Delta t} e^{-\frac{\tilde{b}q}{q't}} dt - Y^*_{m(k)}\Delta t = \tilde{Y}_0 \int_{m_{(k)}}^{m_{(k)}+\Delta t} e^{-\frac{\tilde{b}'}{t}} dt - Y^*_{m(k)}\Delta t, \tag{7}$$

where $\tilde{b}' = \tilde{b} \cdot q / q'$ [8]. In the previous example, originally the capacity allocated for making the product is about 5000 wafers per month, i.e. q = 5000. To enhance the competitiveness, re-allocating an additional capacity of 3000 wafers per month to the product is being considered. The new mid-term competitiveness at each check point is evaluated as

$$\tilde{c}'_{(1)} = (-0.21, 0.03, 0.26);\ \tilde{c}'_{(2)} = (-0.24, -0.00, 0.23);\ \tilde{c}'_{(3)} = (-0.36, -0.12, 0.11).$$

The new long-term competitiveness becomes

1. The minimal mid-term competitiveness = (-0.36, -0.12, 0.11).
2. The average mid-term competitiveness = (-0.27, -0.03, 0.2).
3. The weighted average competitiveness = (-0.31, -0.05, 0.20).

These results also fall in the category “SC”. Because they are very small, to compare with the original values, the percentage of improvement in the defuzzified result is an appropriate index:

$$\frac{d(\text{the new competitiveness}) - d(\text{the original value})}{|\, d(\text{the original value}) \,|} \cdot 100\% , \tag{8}$$

where the center-of-gravity (COG) formula [17] is applied for defuzzification. The results are shown in Table 2. The effects are significant.

Table 2. The percentage of improvement

Measure	Percentage of Improvement
The minimal competitiveness	8%
The average competitiveness	25%
The weighted average competitiveness	16%

In the previous example, after re-allocating an additional capacity of 3000 wafers per month to the product, the trend in the mid-term competitiveness remains unchanged, which implies that if the entire product life cycle is observed, then capacity re-allocation did not take effect because the trend in the competitiveness is not easy to change. That conforms to the viewpoint of many experts in this field – It is very difficult to improve the competitiveness of a product merely with capacity re-allocation. To optimize the performance of capacity re-allocation on the long-term competitiveness, the following fuzzy nonlinear programming model is constructed:

$$\text{Max } \tilde{R} = (\sum_{\text{all } k} m_{(k)}\tilde{c}'_{(k)}(-)K\overline{m}\overline{\tilde{c}}') \Big/ \sqrt{\sum_{\text{all } k} m_{(k)}^2 - K\overline{m}^2} \sqrt{\sum_{\text{all } k} \tilde{c}'^2_{(k)} (-)K\overline{\tilde{c}}'^2} \tag{9}$$

s.t.

$$\tilde{c}'_{(k)} = \tilde{Y}_0 \int_{m_{(k)}}^{m_{(k)}+\Delta t} e^{-\frac{\tilde{b}q}{q't}} dt - Y^*_{m(k)}\Delta t, \tag{10}$$

$$\overline{\tilde{c}}' = \sum_{k=1}^{K} \tilde{c}'_{(k)} \Big/ K, \tag{11}$$

$$q' - q \le \Delta q, \tag{12}$$

$$q' - q \ge 0. \tag{13}$$

The FNP model can be converted into an equivalent NP model to be solved with existing optimization software:

$$\text{Max } \sum_{\alpha=0}^{1} \frac{R^L(\alpha) + R^R(\alpha)}{2} / 11 \tag{14}$$

s.t.

$$R^L(\alpha) = \min_{\text{all } P,Q} (\frac{\sum_{k\in P} m_{(k)}c'^L_{(k)}(\alpha) + \sum_{k\in Q} m_{(k)}c'^R_{(k)}(\alpha) - \overline{m}(\sum_{k\in P} c'^L_{(k)}(\alpha) + \sum_{k\in Q} c'^R_{(k)}(\alpha))}{\sqrt{\sum_{\text{all } k} m_{(k)}^2 - K\overline{m}^2} \sqrt{\sum_{k\in P} c'^L_{(k)}(\alpha)^2 + \sum_{k\in Q} c'^R_{(k)}(\alpha)^2 - (\sum_{k\in P} c'^L_{(k)}(\alpha) + \sum_{k\in Q} c'^R_{(k)}(\alpha))^2 / K}}) \tag{15}$$

$$R^R(\alpha) = \max_{\text{all } P,Q} (\frac{\sum_{k\in P} m_{(k)}c'^L_{(k)}(\alpha) + \sum_{k\in Q} m_{(k)}c'^R_{(k)}(\alpha) - \overline{m}(\sum_{k\in P} c'^L_{(k)}(\alpha) + \sum_{k\in Q} c'^R_{(k)}(\alpha))}{\sqrt{\sum_{\text{all } k} m_{(k)}^2 - K\overline{m}^2} \sqrt{\sum_{k\in P} c'^L_{(k)}(\alpha)^2 + \sum_{k\in Q} c'^R_{(k)}(\alpha)^2 - (\sum_{k\in P} c'^L_{(k)}(\alpha) + \sum_{k\in Q} c'^R_{(k)}(\alpha))^2 / K}}) \tag{16}$$

$$
\begin{aligned}
c'^{L}_{(k)}(\alpha) = Y_0^L(\alpha)(&-5\cdot 10^{-8}\frac{(m_{(k)}+\Delta t)^4}{b'^L(\alpha)^3} + 1.67\cdot 10^{-5}\frac{(m_{(k)}+\Delta t)^3}{b'^L(\alpha)^2} \\
&-0.0021\frac{(m_{(k)}+\Delta t)^2}{b'^L(\alpha)} + 1.1205(m_{(k)}+\Delta t) + 5\cdot 10^{-8}\frac{m_{(k)}^4}{b'^L(\alpha)^3} \\
&-1.67\cdot 10^{-5}\frac{m_{(k)}^3}{b'^L(\alpha)^2} + 0.0021\frac{m_{(k)}^2}{b'^L(\alpha)} - 1.1205 m_{(k)}) - Y^*_{m(k)}\Delta t
\end{aligned} \tag{17}
$$

$$
\begin{aligned}
c'^{R}_{(k)}(\alpha) = Y_0^R(\alpha)(&-5\cdot 10^{-8}\frac{(m_{(k)}+\Delta t)^4}{b'^R(\alpha)^3} + 1.67\cdot 10^{-5}\frac{(m_{(k)}+\Delta t)^3}{b'^R(\alpha)^2} \\
&-0.0021\frac{(m_{(k)}+\Delta t)^2}{b'^R(\alpha)} + 1.1205(m_{(k)}+\Delta t) + 5\cdot 10^{-8}\frac{m_{(k)}^4}{b'^R(\alpha)^3} \\
&-1.67\cdot 10^{-5}\frac{m_{(k)}^3}{b'^R(\alpha)^2} + 0.0021\frac{m_{(k)}^2}{b'^R(\alpha)} - 1.1205 m_{(k)}) - Y^*_{m(k)}\Delta t
\end{aligned} \tag{18}
$$

$$
b'^L(\alpha) = b^L(\alpha)\cdot q/q' \tag{19}
$$

In the previous example, the original capacity q = 5000, while the optimal solution is q' = 13353.

5 Conclusions

Chen's approach is extended in this study to evaluate the long-term competitiveness of a product based on its yield learning model in various ways. For this purpose, a longer time period than that adopted in Chen's study involving multiple check points is observed. Subsequently, to enhance the long-term competitiveness of the product, capacity re-allocation is shown to be helpful. The effects are modeled with a FNP model. To demonstrate the practicability of the proposed methodology, a detailed example using the practical data from a real semiconductor manufacturing factory is adopted. This study was financially supported by the National Science Council of Taiwan.

References

1. Armstrong, E.: Principles for competitive semiconductor manufacturing. In: Proceedings of IEEE SEMI International Semiconductor Manufacturing Science Symposium 1989, pp. 67–72 (1989)
2. Chen, T.: A fuzzy logic approach for incorporating the effects of managerial actions on semiconductor yield learning. In: Proceedings of 2007 International Conference on Machine Learning and Cybernetics, Hong Kong, pp. 1979–1984 (2007)
3. Chen, T.: Evaluating the mid-term competitiveness of a product in a semiconductor fabrication factory with a systematic procedure. Computers & Industrial Engineering (2007) (in press)

4. Chen, T.: A FNP approach for evaluating and enhancing the long-term competitiveness of a semiconductor fabrication factory through yield learning modeling. International Journal of Advanced Manufacturing Technology (2008) (in press)
5. Chen, T.: A fuzzy back propagation network for output time prediction in a wafer fab. Applied Soft Computing 2(3), 211–222 (2003)
6. Chen, T., Wang, M.-J.J.: A fuzzy set approach for yield learning modeling in wafer manufacturing. IEEE Transactions on Semiconductor Manufacturing 12(2), 252–258 (1999)
7. Chen, T., Wang, M.-J.J.: Applying fuzzy set approach to signal detection theory. Fuzzy Sets and Systems 72, 39–49 (1995)
8. Chen, T., Wang, Y.C.: A fuzzy set approach for evaluating and enhancing the mid-term competitiveness of a semiconductor factory. Fuzzy Sets and Systems (2008) (in press)
9. Cunningham, J.A.: The use and evaluation of yield models in integrated circuit manufacturing. IEEE Transactions on Semiconductor Manufacturing 3(2), 60–71 (1990)
10. Gruber, H.: Learning and Strategic Product Innovation: Theory and Evidence for the Semiconductor Industry, The Netherlands. Elsevier Science B. V., Amsterdam (1984)
11. Haller, M., Peikert, A., Thoma, J.: Cycle time management during production ramp-up. Robotics and Computer Integrated Manufacturing 19, 183–188 (2003)
12. Hsieh, K.L., Tong, L.I.: Manufacturing performance evaluation for IC products. International Journal of Advanced Manufacturing Technology 28, 610–617 (2006)
13. Jenkins, T., Phail, F., Sackman, S.: Semiconductor competitiveness in the 1990s. In: Proceedings - Society of Automotive Engineers P223, pp. 249–255 (1990)
14. Li, T.-S., Huang, C.-L., Wu, Z.-Y.: Data mining using genetic programming for construction of a semiconductor manufacturing yield rate prediction system. International Journal of Advanced Manufacturing Technology 17, 355–361 (2006)
15. Liao, S.-H., Hu, T.-C.: Knowledge transfer and competitive advantage on environmental uncertainty: An empirical study of the Taiwan semiconductor industry. Technovation 27(6-7), 402–411 (2007)
16. Lin, C.T., Chang, C.W., Chen, C.B.: Relative control philosophy – balance and continual change for forecasting abnormal quality characteristics in a silicon wafer slicing process. International Journal of Advanced Manufacturing Technology 26, 1109–1114 (2006)
17. Liu, X.: Parameterized defuzzification with maximum entropy weighting function - another view of the weighting function expectation method. Mathematical and Computer Modelling 45, 177–188 (2007)
18. Majd, S., Pindyck, R.S.: The learning curve and optimal production under uncertainty. Rand J. Economics 20(3), 331–343 (1989)
19. Mazzola, J.B., McCardle, K.F.: A Bayesian approach to managing learning-curve uncertainty. Management Science 42(5), 680–692 (1995)
20. Peng, C.-Y., Chien, C.-F.: Data value development to enhance competitive advantage: A retrospective study of EDA systems for semiconductor fabrication. International Journal of Services, Technology and Management 4(4-6), 365–383 (2003)
21. Shimada, Y., Sakurai, K.: A new accurate yield prediction method for system-LSI embedded memories. IEEE Transactions on Semiconductor Manufacturing 16(3), 436–445 (2003)
22. Spence, A.M.: The learning curve and competition. Bell J. Economics 12, 49–70 (1981)
23. Tong, L.-I., Wang, C.H., Chen, D.L.: Development of a new cluster index for wafer defects. International Journal of Advanced Manufacturing Technology 31, 705–715 (2007)
24. Watada, J., Tanaka, H., Shimomura, T.: Identification of learning curve based on possibilistic concepts. In: Applications of Fuzzy Set Theory in Human Factors, The Netherlands. Elsevier Science Publishers B. V., Amsterdam (1986)

Realization of XOR by SIRMs Connected Fuzzy Inference Method

Hirosato Seki[1,2], Satoshi Watanabe[1], Hiroaki Ishii[1], and Masaharu Mizumoto[3]

[1] Osaka University, Suita, Osaka, Japan
[2] Japan Society for the Promotion of Science (JSPS), Japan
[3] Osaka Electro-Communication University, Neyagawa, Osaka, Japan

Abstract. The single input rule modules connected fuzzy inference method (SIRMs method) by Yubazaki et al. can decrease the number of fuzzy rules drastically in comparison with the conventional fuzzy inference methods. Moreover, Seki et al. have proposed a functional type single input rule modules connected fuzzy inference method which generalizes the consequent part of the SIRMs method to function. However, these SIRMs method can not realize XOR (Exclusive OR). In this paper, we propose a "neural network-type SIRMs method" which unites the neural network and SIRMs method, and show that this method can realize XOR. Further, a learning algorithm of the proposed SIRMs method is derived by steepest descent method, and is shown to be superior to the conventional SIRMs method and neural network by applying to identification of nonlinear functions.

Keywords: Fuzzy Inference, SIRMs connected fuzzy inference method, neural network, XOR.

1 Introduction

As for the "IF-THEN" rules in the traditional fuzzy inference methods, all the input items of the system are set to the antecedent part, and all output items are set to the consequent part. Therefore, the problem is apparent that the number of fuzzy rules becomes increasingly huge; hence, the setup and adjustment of fuzzy rules become difficult. On the other hand, a single input rule modules connected type fuzzy inference method (SIRMs method) by Yubazaki et al. [1,2,3] which unifies the inference output from fuzzy rule modules of one input type "IF-THEN" form can reduce the number of fuzzy rules drastically. The method has been applied to nonlinear function identification, control of a first order lag system with dead time, orbital pursuit control of a non-restrained object, and stabilization control of a handstand system etc., and good results are obtained. However, since the number of rules of the SIRMs method is limited compared to the traditional inference methods, inference results gained by the SIRMs method are simple in general.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 252–261, 2009.

From the above reason, Seki et al. [4] have proposed a "functional-type SIRMs method" whose the consequent parts are generalized to functions from real numbers, and have shown that the functional-type SIRMs mehtod is a special case of the Takagi-Sugeno (T-S) inference method [5]. Moreover, functional-type SIRMs method has been applied to nonliear function identification and a medical dyagnosis system, and good results are obtained.

However, these SIRMs method can not be applied to the realization of exclusive OR (XOR) [4].

In this paper, we propose a "neural network-type SIRMs method" which unifies the SIRMs method and neural network, and show that this method can be applied to XOR. Further, a learning algorithm of the proposed method is derived by steepest descent method, and is shown to be superior to the conventional SIRMs method and neural network by applying to identification of nonlinear functions.

2 Union between SIRMs Connected Fuzzy Inference Method and Neural Network

The researches on union between the fuzzy inference and neural network have been studied in [6]–[8]. In this section, we first explain the SIRMs method [1,2,3] and propose a "neural network-type SIRMs method" which unifies the conventional SIRMs method and neural network.

2.1 SIRMs Connected Fuzzy Inference Method

We briefly review the SIRMs method by Yubazaki et al. [1,2,3]. The SIRMs method has p rule modules, and final inference result by the SIRMs method is obtained by the weighted sum of p inference results from rule modules and p weights. The rule modules of SIRMs method are given as

$$\begin{aligned} \text{Rules-1} &: \{x_1 = A_j^1 \longrightarrow y_1 = y_j^1\}_{j=1}^{m_1} \\ &\vdots \\ \text{Rules-}i &: \{x_i = A_j^i \longrightarrow y_i = y_j^i\}_{j=1}^{m_i} \\ &\vdots \\ \text{Rules-}p &: \{x_p = A_j^p \longrightarrow y_p = y_j^p\}_{j=1}^{m_p} \end{aligned} \tag{1}$$

where Rules-i stands for the ith single input rule module, the ith input item x_i is the sole variable of the antecedent part in the Rules-i, and y_i stands for the variable of its consequent part. A_j^i means the fuzzy set of the jth rule of the Rules-i, y_j^i is real value of consequent part, for $i = 1, 2, \ldots, p$; $j = 1, 2, \ldots, m_i$.

The degree of the antecedent part in the jth rule of Rules-i is obtained by (2) for input x_i^0, and the inference result y_i^0 from Rules-i is given as (3).

$$h_j^i = A_j^i(x_i^0) \tag{2}$$

$$y_i^0 = \frac{\sum_{k=1}^{m_i} h_k^i y_k^i}{\sum_{k=1}^{m_i} h_k^i} \tag{3}$$

Final inference result y^0 of the SIRMs method is given by (4), where importance degree for each input item x_i $(i = 1, 2, \ldots, p)$ is set as w_i. That is,

$$y^0 = \sum_{i=1}^{p} w_i y_i^0 \tag{4}$$

2.2 Neural Network-Type SIRMs Connected Fuzzy Inference Method

In this subsection, we propose a "neural network-type SIRMs method" which unifies the conventional SIRMs method and neural network. The rules of neural network-type SIRMs method are given as (1) like the conventional SIRMs method. Moreover, given input x_i^0 to Rules-i, the degree h_j^i of the antecedent part of the jth rule in Rules-i and inference result y_i^0 from the rule modules are also given as (2) and (3), respectively.

Although final output of the conventional SIRMs method is obtained by the weighted sum of n inference results from rule modules and n weights, the proposed method introduces inner layer before final output by the conventional SIRMs method is obtained. Introducing S inner layers, T_s is obtained as follows.

$$T_s = \sum_{i=1}^{n} v_i^s y_i^0 - \theta_1^s \tag{5}$$

where $s = 1, 2, \ldots, S$, v_i^s is weight for y_i^0 in the sth elemental device, and θ_1^s is threshold value in the sth elemental device.

Output H_s in the sth elemental device which uses sigmoid function is given as follows.

$$H_s = \frac{1}{1 + e^{-T_s}} \tag{6}$$

U is obtained as follows.

$$U = \sum_{s=1}^{S} w_s H_s - \theta_2 \tag{7}$$

where w_i stands for importance degree, and θ_2 means threshold value in inner layer.

The final output y^0 of this inference method is obtained as follows by using (3), (5) $\sim$ (7) and the sigmoid function.

$$y^0 = \frac{1}{1 + e^{-U}} \tag{8}$$

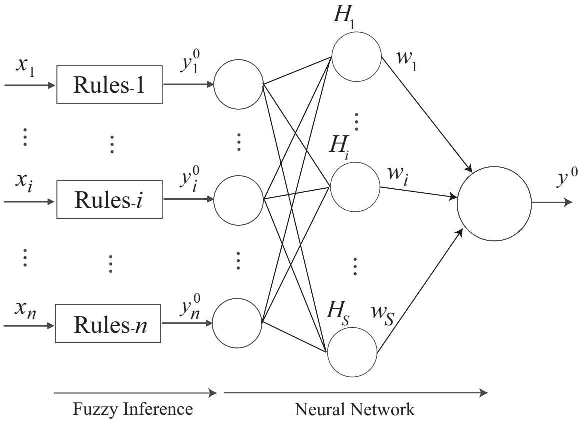


Fig. 1. The structure of the neural network-type SIRMs method

The structure of the neural network-type SIRMs method is illustrated in Fig. 1.

3 Learning Algorithms for Neural Network-Type SIRMs Method

Generally speaking, the setup of the membership functions and fuzzy rules is difficult. Hence, we expect to automatically optimize the membership functions and fuzzy rules based on input-output data in systems. From the reason, several learning algorithms for membership function and fuzzy rules are proposed [9,10]. In this subsection, we propose a learning algorithm for the neural network-type SIRMs method.

Given the training input-output data $(x_{r_1}, x_{r_2}, \ldots, x_{r_m}; y^{T_r})$ for a fuzzy system model, it is usual to use the following objective function E_r for evaluating an error between y^{T_r} and y^{0_r}, which can be regarded as an optimum problem:

$$E_r = \frac{1}{2}(y^{T_r} - y^{0_r})^2 \tag{9}$$

where y^{T_r} is the desired output value, and y^{0_r} the corresponding fuzzy inference result.

In this subsection, triangular type and Gaussian type fuzzy sets are used as two kinds of fuzzy sets. The parameters of center a_j^i, width b_j^i, consequent part y_j^i, and importance degree w_i are obtained by steepest descent method as follows.

(I) Case of triangular type fuzzy sets:

At first, we consider the following triangular type fuzzy set $A_j^i(x_i)$.

$$A_j^i(x_i) = \begin{cases} 1 - |x_i - a_j^i|/b_j^i; \; a_j^i - b_j^i \leq x_i \leq a_j^i + b_j^i \\ 0; \qquad\qquad\qquad\quad \text{otherwise} \end{cases} \tag{10}$$

where a_j^i and b_j^i $(i = 1, 2, \ldots, q; j = 1, 2, \ldots, m_i)$ stand for the center and width, respectively. From (10), the learning algorithms at $t + 1$ step of each parameter are obtained as follows.

$$\begin{aligned} & a_j^i(t+1) \\ & = a_j^i(t) + \alpha \cdot (y^T - y^0(t)) \cdot y^0(t) \cdot (1 - y^0(t)) \cdot \sum_{q=1}^{S} w_q(t) H_q(t)(1 - H_q(t)) v_i^q(t) \\ & \quad \cdot \frac{y_j^i(t) - y_i^0(t)}{\sum_{k=1}^{m_i} h_k^i(t)} \cdot \frac{\mathrm{sgn}(x_i - a_j^i(t))}{b_j^i(t)} \end{aligned} \tag{11}$$

$$\begin{aligned} & b_j^i(t+1) \\ & = b_j^i(t) + \beta \cdot (y^T - y^0(t)) \cdot y^0(t) \cdot (1 - y^0(t)) \cdot \sum_{q=1}^{S} w_q(t) H_q(t)(1 - H_q(t)) v_i^q(t) \\ & \quad \cdot \frac{y_j^i(t) - y_i^0(t)}{\sum_{k=1}^{m_i} h_k^i(t)} \cdot \frac{|xi - a_j^i(t)|}{(b_j^i(t))^2} \end{aligned} \tag{12}$$

$$\begin{aligned} & y_j^i(t+1) \\ & = y_j^i(t) + \gamma \cdot (y^T - y^0(t)) \cdot y^0(t) \cdot (1 - y^0(t)) \cdot \sum_{q=1}^{S} w_q(t) H_q(t)(1 - H_q(t)) v_i^q(t) \\ & \quad \cdot \frac{h_j^i(t)}{\sum_{k=1}^{m_i} h_k^i(t)} \end{aligned} \tag{13}$$

$$\begin{aligned} & w_s(t+1) \\ & = w_s(t) + \delta \cdot (y^T - y^0(t)) \cdot y^0(t) \cdot (1 - y^0(t)) \cdot H_s(t) \end{aligned} \tag{14}$$

$$\begin{aligned} & v_i^s(t+1) \\ & = v_i^s(t) + \epsilon \cdot (y^T - y^0(t)) \cdot y^0(t) \cdot (1 - y^0(t)) \cdot w_s(t) \cdot H_s(t) \cdot (1 - H_s(t)) \\ & \quad \cdot y_i^0(t) \end{aligned} \tag{15}$$

where α, β, γ, δ and ϵ are the learning rates in the learning process, t means the learniing iteration number, and *sgn* stands for the following sign function.

$$\mathrm{sgn}(x) = \begin{cases} 1 : x > 0 \\ 0 : x = 0 \\ -1 : x < 0 \end{cases} \tag{16}$$

(II) Case of Gaussian type fuzzy sets:

We consider the following Gaussian type fuzzy set $A_j^i(x_i)$.

$$A_j^i(x_i) = \exp\left(-\frac{(x_i - a_j^i)^2}{b_j^i}\right) \tag{17}$$

where a_j^i and b_j^i $(i = 1, 2, \ldots, q; j = 1, 2, \ldots, m_i)$ stand for the center and width, respectively. From (17), the learning algorithm of each parameter is obtained as follows.

$$\begin{aligned} & a_j^i(t+1) \\ & = a_j^i(t) + \alpha \cdot (y^T - y^0(t)) \cdot y^0(t) \cdot (1 - y^0(t)) \cdot \sum_{q=1}^{S} w_q(t) H_q(t)(1 - H_q(t)) v_i^q(t) \\ & \quad \cdot \frac{y_j^i(t) - y_i^0(t)}{\displaystyle\sum_{k=1}^{m_i} h_k^i(t)} \cdot \frac{2(x_i - a_j^i(t))}{b_j^i(t)} \cdot h_j^i(t) \end{aligned} \tag{18}$$

$$\begin{aligned} & b_j^i(t+1) \\ & = b_j^i(t) + \beta \cdot (y^T - y^0(t)) \cdot y^0(t) \cdot (1 - y^0(t)) \cdot \sum_{q=1}^{S} w_q(t) H_q(t)(1 - H_q(t)) v_i^q(t) \\ & \quad \cdot \frac{y_j^i(t) - y_i^0(t)}{\displaystyle\sum_{k=1}^{m_i} h_k^i(t)} \cdot \left(\frac{xi - a_j^i(t)}{b_j^i(t)}\right)^2 \cdot h_j^i(t) \end{aligned} \tag{19}$$

$$\begin{aligned} & y_j^i(t+1) \\ & = y_j^i(t) + \gamma \cdot (y^T - y^0(t)) \cdot y^0(t) \cdot (1 - y^0(t)) \cdot \sum_{q=1}^{S} w_q(t) H_q(t)(1 - H_q(t)) v_i^q(t) \\ & \quad \cdot \frac{h_j^i(t)}{\displaystyle\sum_{k=1}^{m_i} h_k^i(t)} \end{aligned} \tag{20}$$

$$\begin{aligned} & w_s(t+1) \\ & = w_s(t) + \delta \cdot (y^T - y^0(t)) \cdot y^0(t) \cdot (1 - y^0(t)) \cdot H_s(t) \end{aligned} \tag{21}$$

$$\begin{aligned} & v_i^s(t+1) \\ & = v_i^s(t) + \epsilon \cdot (y^T - y^0(t)) \cdot y^0(t) \cdot (1 - y^0(t)) \cdot w_s(t) \cdot H_s(t) \cdot (1 - H_s(t)) \\ & \quad \cdot y_i^0(t) \end{aligned} \tag{22}$$

where α, β, γ, δ, ϵ and t have the same meanings as the case of triangular type fuzzy sets.

4 Identification for XOR by Neural Network-Type SIRMs Method

In identifying XOR (exclusive OR), we use five membership functions for the inputs x_1 and x_2, where the centers of membership functions $A_1^i, A_2^i, \ldots, A_5^i$ for $i = 1, 2$ are -1, -0.5, 0, 0.5 and 1, and each width of membership functions is assumed to be 0.5.

Table 1 shows the training data for XOR, where 441 checking data (x_1, x_2) are employed from (0, 0) to (1, 1), as shown in Table 2.

Here, we obtain the error of evaluation regarding the desired model and inference model, where the error of evaluation is mean square error for checking data. Here, the learning rates are taken as $\alpha = 0.001$, $\beta = 0.0001$, $\gamma = 0.01$, $\delta = 0.01$.

The inputs of the neural network-type SIRMs method (NN-SIRMs, for short in the table) and conventional SIRMs method are used in Table 1. The inference results are shown in Table 3 after the learning iterations are executed 1000 times.

Moreover, the inference results of the conventional SIRMs method and proposed method are shown in Fig. 2.

Table 1. The training data for XOR

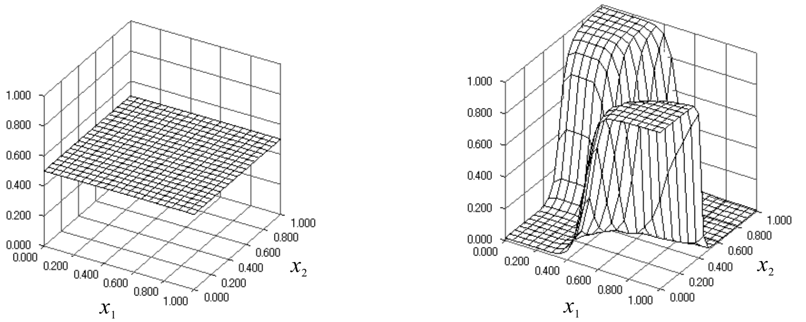
$\boldsymbol{X}$	x_1	x_2	y
$\boldsymbol{d}_1$	0	0	0
$\boldsymbol{d}_2$	0	1	1
$\boldsymbol{d}_3$	1	0	1
$\boldsymbol{d}_4$	1	1	0

Table 2. The checking data

$\boldsymbol{X}$	x_1	x_2
$\boldsymbol{d}_1$	0	0
$\boldsymbol{d}_2$	0	0.05
$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{d}_{20}$	0	0.95
$\boldsymbol{d}_{21}$	0	1
$\boldsymbol{d}_{22}$	0.05	0
$\boldsymbol{d}_{23}$	0.05	0.05
$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{d}_{440}$	1	0.95
$\boldsymbol{d}_{441}$	1	1

Table 3. The error of evaluation for XOR

	SIRMs	NN-SIRMs
d_1	0.5	1.60×10^{-2}
d_2	0.5	1.56×10^{-2}
d_3	0.5	1.23×10^{-2}
d_4	0.5	0.77×10^{-2}



(a) Conventional SIRMs method

(b) Neural Network type SIRMs method

Fig. 2. The inference results of each method for XOR

From the above results, we clarify that the proposed method can obtain good results for identifying XOR, though the conventional SIRMs method can not realize XOR.

5 Identification of Nonlinear Functions by Neural Network-Type SIRMs Method

In this subsection, we apply the neural network-type SIRMs method and the above learning algorithm to the following two nonlinear functions with two inputs variables and one output variable in order to compare them with the conventional SIRMs method and neural network for identifying and evaluating the problems, and show the effectiveness of the proposal method.

The following two nonlinear functions are used:

$$\text{Function1.}\ \ y = (2x_1 + 4x_2^2 + 0.1)^2/37.21 \tag{23}$$

$$\text{Function2.}\ \ y = (2\sin(\pi x_1) + \cos(\pi x_2) + 3)/6 \tag{24}$$

where $x_1,\ x_2 \in [-1,\ 1]$ are input variavle, and $y \in [0,\ 1]$ is a normalized output variable.

In order to identifying nonlinear functions, we use five membership functions for the inputs x_1 and x_2, where the centers of the membership functions $A_1^i, A_2^i, \ldots, A_5^i$ for $i = 1,\ 2$ are -1, -0.5, 0, 0.5 and 1, and each width of membership functions is 0.5.

We obtain the error of evaluation regarding desired model and inference model in previous section.

In our case, 2601 checking data $(x_1,\ x_2)$ are employed from $(-1, -1)$ to $(1, 1)$, and 49 training data are used from 2601 checking data in a random order. For these data, learning iterations are executed 1000 times, where the learning rates are $\alpha = 0.001,\ \beta = 0.0001,\ \gamma = 0.01,\ \delta = 0.01$.

In the following, we identify Functions 1 and 2 by using the proposal method and conventional SIRMs method in the case of triangular type and Gaussian type membership functions, respectively.

Table 4. The error of evaluation for Function 1 of (23)

	Triangular type		Gaussian type		
Case	SIRMs	NN-SIRMs	SIRMs	NN-SIRMs	NN
1	0.0849	0.0312	0.0889	0.0289	0.1264
2	0.0687	0.0351	0.0776	0.0344	0.1126
3	0.0733	0.0280	0.0900	0.0296	0.1261
4	0.0701	0.0320	0.0802	0.0295	0.1140
5	0.0884	0.0365	0.0896	0.0320	0.1226
6	0.0677	0.0370	0.0688	0.0339	0.1086
7	0.0675	0.0376	0.0705	0.0452	0.1059
8	0.0728	0.0269	0.0791	0.0274	0.1181
9	0.0614	0.0291	0.0702	0.0276	0.1070
10	0.0799	0.0265	0.0845	0.0274	0.1159
Average	0.0735	0.0320	0.0799	0.0316	0.1157

Table 5. The error of evaluation for Function 2 of (24)

	Triangular type		Gaussian type		
Case	SIRMs	NN-SIRMs	SIRMs	NN-SIRMs	NN
1	0.0354	0.0271	0.0302	0.0295	0.1445
2	0.0357	0.0333	0.0296	0.0280	0.1370
3	0.0335	0.0279	0.0263	0.0274	0.1344
4	0.0375	0.0312	0.0299	0.0295	0.1328
5	0.0354	0.0287	0.0281	0.0279	0.1401
6	0.0383	0.0329	0.0301	0.0289	0.1347
7	0.0349	0.0298	0.0277	0.0284	0.1309
8	0.0348	0.0282	0.0291	0.0272	0.1422
9	0.0349	0.0300	0.0279	0.0282	0.1313
10	0.0329	0.0305	0.0276	0.0290	0.1335
Average	0.0353	0.0300	0.0287	0.0284	0.1361

For the nonlinear function 1, learninig iterations are executed 1000 times, and 10 simulations are run. Table 4 shows the error of evaluation using the checking data for identifying Function 1. Here, the error of evaluation denotes a mean square error for the checking data.

For the nonlinear function 2, learninig iterations are executed 1000 times, and 10 simulations are run. Table 5 shows the error of evaluation using the checking data for identifying Function 2.

All methods give good results regarding Function 2 as an additive function, as shown in Table 5. Moreover, although the conventional SIRMs method and neural network do not necessarily obtain good results for the Function 1 as a multiplicative function, the neural network-type SIRMs method has good results, as shown in Table 4. From this results, we have clarified that the neural network-type SIRMs method can be applied to multiplicative functions as well as additive functions.

Therefore, we have shown that the proposed method is superior to the conventional SIRMs method and neural network.

6 Conclusion

In order to enhance the SIRMs method, in this paper, we have proposed the "neural network-type SIRMs method" which unifies the conventional SIRMs method and neural network and shown that the proposed SIRMs method can realize XOR. Further, a learning algorithm of the neural network-type SIRMs method are derived by steepest descent method, and is shown to be superior to the conventional method by applying to identification of nonlinear functions.

Further studies are required to compare the neural netwrok-type SIRMs method, conventional neural network and adaptive neuro-fuzzy inference systems (ANFIS).

Acknowledgment

This work was partially supported by a Grant-in-Aid for Scientific Research (Project No. 20·797) from the Japan Society for the Promotion Science (JSPS).

References

1. Yubazaki, N., Yi, J., Otani, M., Hirota, K.: SIRMs dynamically connected fuzzy inference model and its applications. In: Proc. IFSA 1997, vol. 3, pp. 410–415 (1997)
2. Yi, J., Yubazaki, N., Hirota, K.: A proposal of SIRMs dynamically connected fuzzy inference model for plural input fuzzy control. Fuzzy Sets Syst. 125, 79–92 (2002)
3. Yi, J., Yubazaki, N., Hirota, K.: A new fuzzy controller for stabilization of parallel-type double inverted pendulum system. Fuzzy Sets Syst. 126, 105–119 (2002)
4. Seki, H., Ishii, H., Mizumoto, M.: On the generalization of single input rule modules connected type fuzzy reasoning method. IEEE Trans. Fuzzy Syst. 16, 1180–1187 (2008)
5. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst. Man Cybern. 15, 116–132 (1985)
6. Jang, J.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans. Syst., Man Cybern. 23, 665–685 (1993)
7. Wang, J.S., George, C.S.: Self-adaptive neuro-fuzzy inference systems for classification applications. IEEE Trans. Fuzzy Syst. 10, 790–802 (2002)
8. Iyatomi, H., Hagiwara, M.: Adaptive fuzzy inference neural network. Pattern Recognition 37, 2049–2057 (2004)
9. Juang, C.F.: A TSK type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms. IEEE Trans. Fuzzy Syst. 10, 155–170 (2002)
10. Yu, W., Li, X.: Fuzzy identification using fuzzy neural networks with stable learning algorithms. IEEE Trans. Fuzzy Syst. 12, 411–420 (2004)

A Robust Support Vector Regression Based on Fuzzy Clustering

Horng-Lin Shieh

Department of Electrical Engineering, Saint John's University,
499, Sec. 4, TamKing Rd., Taipei, TamSui 25135, Taiwan
shieh@mail.sju.edu.tw

Abstract. Support Vector Regression (SVR) has been very successful in pattern recognition, text categorization and function approximation. In real application systems, data domain often suffers from noise and outliers. When there is noise and/or outliers existing in sampling data, the SVR may try to fit those improper data and obtained systems may have the phenomenon of *overfitting*. In addition, the memory space for storing the kernel matrix of SVR will be increment with O (N^2), where N is the number of training data. In this paper, a robust support vector regression is proposed for nonlinear function approximation problems with noise and outliers.

Keywords: Support vector regression, outlier, noise, robust, Fuzzy Clustering.

1 Introduction

The theory of support vector machines (SVM) developed by Vapnik [1] in 1995 is gaining popularity due to many attractive features. It has been successfully applied to a number of applications such as classification, time prediction, pattern recognition and regression [2]~[7]. The SVM can also be applied to regression problems by introduction of an alternative loss function [2]. Such approaches are often called the support vector regression (SVR).

SVM adopted the quadratic programming (QP) to maximize the margin. The computing task became very challenge when the number of data was beyond a few thousands [8]. For example, within Fig. 1, there are 500 sampling data generated from a *sin* wave with Gaussian noise N (0, 0.447). The SVR algorithm is adopted to construct this function. The entries of kernel matrix of SVR are floating-point number, and each floating-point number needs 4 bytes for storing. So, the totally memory requirement is 1000000 bytes. The SVR algorithm is carried out on a Pentium 4, 1.8GHz with 128MB of memory and running with Windows XP. The totally execution time of the simulation is 21941 seconds (above 6 hours).

In [9], the authors proposed a generalized decomposition strategy for the standard SVM. In [10], Lee and Huang proposed to restrict the number of support vectors by solving the reduced support vector machines (RSVM). In RSVM the smaller matrix can be stored in memory, so optimization algorithms such as Newton method can be applied [11]. In [12], a reduced set method based on the measurement of similarity between samples is developed. In this paper, these samples are so similar

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 262–270, 2009.

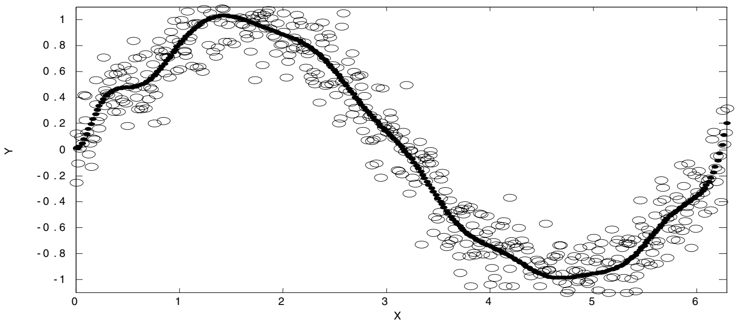


Fig. 1. Sin wave with noise N(0, 0.447)

to the special data point that their influence on the prediction function can be ignored. However, like the K-means clustering algorithm, the disadvantage of this algorithm is the number of clusters must be predetermined. But in some real applications, there is no information to predefine the number of the clusters.

In real applications, data is bound to have noise and outliers. In system modeling when there is noise and/or outliers exist in sampling data, the system models may try to fit those improper data and obtained systems may have the phenomenon of overfitting [13].

In this paper, in order to overcome above problems, a robust fuzzy clustering method is proposed to greatly mitigate the influence of noise and outliers in sampling data and then SVR method is used to construct the system models.

2 Support Vector Regression

Let the sampling data be represented as $\{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in R^d, y_i \in \{-1,1\}\}$, i=1,2, … n. In SVR method, the regression function is approximated by the following function as

$$f = \sum_{i=1}^{n} \mathbf{w}_i \boldsymbol{\varphi}(\mathbf{x}_i) + b \ , \tag{1}$$

where $\{\boldsymbol{\varphi}(\mathbf{x}_i)\}_{i=1}^{n}$ are the features of inputs, $\{\mathbf{w}_i\}_{i=1}^{n}$ and b are coefficients. The coefficients are estimated by minimizing the regularized risk function [12]

$$R(C) = C\frac{1}{n}\sum_{i=1}^{n} L(y, f) + \frac{1}{2} \| \mathbf{w} \|^2 , \tag{2}$$

where C is a regular constant and $L(y,f)$ adopts the ε -insensitive loss function defined as following:

$$L(y, f) = \begin{cases} | y - f | - \varepsilon, & | y - f | \geq \varepsilon, \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

and $\varepsilon \geq 0$ is a predefined parameter.

SVR introduces the slack variables ξ,ξ^* and leads equation (2) to the following constrained function

$$\text{Mininize} \quad R(\mathbf{w},\xi^*) = C^*\sum_{i=1}^{n}(\xi_i+\xi_i^*)+\frac{1}{2}\|\mathbf{w}\|^2 , \tag{4}$$

subject to

$$\mathbf{w}\boldsymbol{\varphi}(\mathbf{x}_i)+b-y_i \le \varepsilon+\xi_i^*,\ y_i-\mathbf{w}\boldsymbol{\varphi}(x_i)-b_i \le \varepsilon+\xi_i,\quad \xi,\xi^* \ge 0 , \tag{5}$$

where ξ,ξ^* are slack variables representing upper and lower constraints on the outputs of the system. Thus, equation (1) becomes the explicit form

$$f(\mathbf{x},\alpha_i,\alpha_i^*) = \sum_{i=1}^{n}\mathbf{w}_i\boldsymbol{\varphi}(\mathbf{x}_i)+b = \sum_{i=1}^{n}(\alpha_i-\alpha_i^*)\boldsymbol{\varphi}(\mathbf{x}_i)^T\boldsymbol{\varphi}(\mathbf{x}_i)+b \ . \tag{6}$$

By using the Lagrange multiplier method, the minimization of equation (4) leads to the following dual optimization problem [14].

$$\text{Mininize} \quad Q(\alpha_i,\alpha_i^*) = \varepsilon\sum_{i=1}^{n}(\alpha_i+\alpha_i^*)-\sum_{i=1}^{n}y_i(\alpha_i^*-\alpha_i)$$

$$+\frac{1}{2}\sum_{i,s=1}^{n}(\alpha_i^*-\alpha_i)(\alpha_s^*-\alpha_s)\boldsymbol{\varphi}(\mathbf{x}_i)^T\boldsymbol{\varphi}(\mathbf{x}_i) \ , \tag{7}$$

subject to the constraint:

$$\sum_{i=1}^{n}\alpha_i^* = \sum_{i=1}^{n}\alpha_i \ , 0<\alpha_i,\alpha_i^*<C ,\quad i=1,\ldots,n. \tag{8}$$

In equation (7), the inner product of functions $\boldsymbol{\varphi}(\mathbf{x}_i)$ can replaced by a so-called kernel function with the form

$$K(\mathbf{x}_i,\mathbf{x}_j) = \boldsymbol{\varphi}(\mathbf{x}_i)^T\boldsymbol{\varphi}(\mathbf{x}_j) \ . \tag{9}$$

It was shown in [1] that the solution of SVR approach is in the form of the following linear expansion of kernel functions:

$$f(\mathbf{x},\alpha,\alpha^*) = \sum_{i=1}^{n}(\alpha_i^*-\alpha_i)K(\mathbf{x}_i,\mathbf{x}_j)+b \ . \tag{10}$$

In this paper, the Gaussian function is used as the kernel function. Hence, equation (10) can be rewritten as [14]

$$f(\mathbf{x}) = \sum_{i=1}^{SV}w_i\exp\left\{-\|\mathbf{x}-\mathbf{x}_i\|^2 \Big/ 2\sigma^2\right\}+b \ , \tag{11}$$

where SV is the number of SVs, $w_i=(\alpha_i^*-\alpha_i)\neq 0$ and $\mathbf{x}$ are SV's.

3 Data Fuzzy Clustering Based on Distance Relation

Let v_i and v_j be represented the centers of cluster i and j respectively, the distance relation between centers of two clusters is defined by the following equation

$$r_{ij} = \exp(-\frac{\left\|v_i - v_j\right\|^2}{2\sigma^2}), \qquad i = 1, 2, \ldots, n, \qquad j = 1, 2, \ldots, n \ , \tag{12}$$

where $\|v_i - v_j\|$ represents the Euclidean distance between v_i and v_j , and σ is the width of the Gaussian function. Initially, each data point is formed one cluster containing the data point itself. Clusters are then to be merged into larger clusters based on how separated among clusters, which, in other words, on how separated among cluster centers.

The center $v_i^{'}$ of the newly combined cluster is defined by equation (13)

$$v_i^{'} = \frac{\sum_{j=1}^{n} r_{ij}^m v_j}{\sum_{j=1}^{n} r_{ij}^m}, \qquad j = 1, 2, ..., n, \quad m \geq 1 , \tag{13}$$

where m is a factor that weighs the importance of the distance relation r_{ij}. By equation (13), the referenced cluster center v_i is replaced by the weighted-average of those cluster centers with high distance relations to the referenced center.

4 The Robust FCM Method

Set X={x$_1$, x$_2$, …, x$_n$} $\subset \Re^d$ be a data set with noise and outliers in a d-dimensional feature space, and v={ v_1, v_2, …, v_c } be a cluster center set, where $v_k \in \Re^d$. Suppose v_{i-1} ,v_i ,v_{i+1} in Fig. 2 are three neighboring local cluster centers generated by the distance relation in Section 3. If a new point v_i^* is positioned in a location inside the triangle formed by v_{i-1}, v_i, v_{i+1} as shown in Fig. 2, then the system model formed by v_{i-1}, v_i^* , v_{i+1} would be smoother. By the rule of triangle, the sum of distances between v_{i-1}, v_i and v_i, v_{i+1}, is greater than the sum of distances between v_{i-1}, v_i^* and v_i^* , v_{i+1}. This relation can be represented by equation (14) [15]

$$\| v_i - v_{i-1} \|^2 + \| v_{i+1} - v_i \|^2 > \| v_i^* - v_{i-1} \|^2 + \| v_{i+1} - v_i^* \|^2 \ . \tag{14}$$

For a cluster i with center v_i, the sum of distances between v_i and the data points, say x_{ki} (k = 1, 2, …, m_i), in the cluster is minimized according to the definition of a cluster center. Consequently, when v_i is replaced by v_i^* , it is natural that the sum of

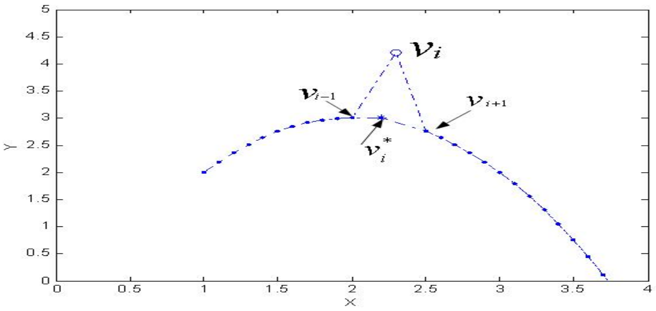


Fig. 2. An example of non-smooth curve

distances between data points x_{ki} (k = 1, 2, …, m_i) and v_i^* will be greater than the sum of distances between these data points and v_i. This relation is represented as follows

$$\sum_{k=1}^{m_i} \| x_{ki} - v_i^* \| \geq \sum_{k=1}^{m_i} \| x_{ki} - v_i \| \ . \tag{15}$$

Equations (14) and (15) can be combined using Lagrange multiplier as equation (16) to find the optimal solution of the two terms where v_i^* is being replaced by v_k for simpler expression.

$J_{RFCM}(\mu, v) =$

$$\sum_{i=1}^{n}\sum_{k=1}^{c}(\mu_{ik})^m \| x_i - v_k \|^2 + \alpha(\| v_1 - v_2 \|^2 + \sum_{k=2}^{c-1}(\| v_k - v_{k-1} \|^2 + \| v_{k+1} - v_k \|^2) + \| v_c - \mathrm{v}_{c\text{-}1} \|^2)$$

$$= \sum_{i=1}^{n}\sum_{k=1}^{c}(\mu_{ik})^m \| x_i - v_k \|^2 + \lambda \sum_{k=1}^{c-1} \| v_{k+1} - v_k \|^2 \text{ , where } \lambda = 2\alpha \ , \tag{16}$$

and λ is the Lagrange multiplier. To find the optimal solution of equation (16), the derivative of $J_{RFCM}(\mu, v)$ with respect to v_k is set to zero while fixing μ . Equation (17) expresses this relationship

$$\frac{\partial J_{RFCM}(\mu, v)}{\partial v_k} = 0 = \sum_{i=1}^{n} \mu_{ik}^m (x_i - v_k) + \lambda(v_{k+1} - v_k) \ . \tag{17}$$

The value of v_k can be obtained as equation (18)

$$v_k = \frac{\lambda v_{k+1} + \sum_{i=1}^{n} \mu_{ik}^m x_i}{\lambda + \sum_{i=1}^{n} u_{ik}^m} \ . \tag{18}$$

To indicate the fact that a smaller distance between cluster center v_k and data point x_i should result in larger value of membership grade, the membership function of the distance is defined as a Gaussian in equation (19)

$$\mu_{ik} = \exp(-\frac{\| x_i - v_k \|^2}{2\sigma^2}), \tag{19}$$

where μ_{ik} is the membership grade of data point x_i belonging to class k, v_k is the center of cluster k, $\|x_i\text{-}v_k\|$ represents the Euclidean distance between x_i and v_k , and σ is the width of the Gaussian function. The RFCM is stated as in Algorithm 1.

Algorithm 1. RFCM algorithm

```
Step 1: Set value of $m$ satisfying $1<m<\infty$ and allowed maximum number of
        iteration.
Step 2: Calculate initial cluster centers $v_k$, $k=1, 2, \ldots$ , c by equations (12) (13) in
        section 3.
Step 3: Calculate the membership matrix of $\mu$ by equation (19).
Step 4: Update fuzzy cluster centers using equation (18).
Step 5: Calculate objective function value by equation (16). Stop if either the
        value is below a certain tolerance value or its improvement over
        previous iteration is below a certain threshold or the number of iteration
        is over the maximum value.
Step 6: Calculate the new $\mu$ using equation (19). Go to Step 4.
```

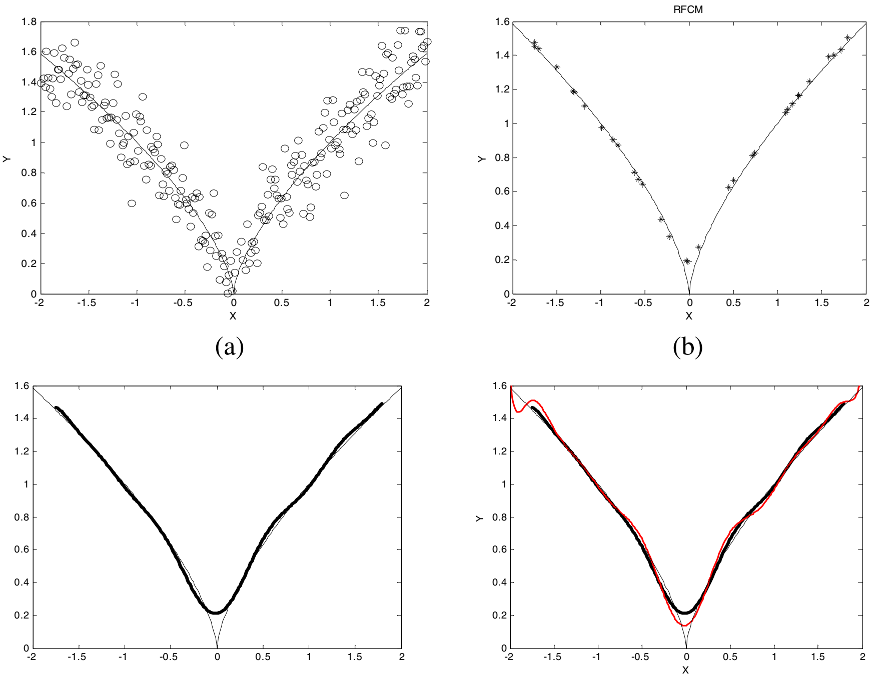
The proposed method of this paper is first to use a distance-relation analysis to search out initial clusters of sampling data. Then, the RFCM is used to mitigate the influence of data noise and outliers. The resultant clusters of the RFCM are then used for SVR training. For a large-scale training data set with noise and outlier, the first step can reduce the size of training data. The RFCM can mitigate the influence of data noise and outliers.

5 Experimental Results

In this section, two experiments are illustrated and their results have shown the proposed approach has better performance and less execution time than original SVR method in various kinds of data domains with data noise and outliers. All simulations are carried out on a Pentium 4, 1.8GHz with 128MB of memory and running with Windows XP.

Example 1
In this example, a function used in [16] is defined as

$$y = x^{\frac{2}{3}}, \text{ with x} \in [-2,2]. \tag{20}$$



(a) (b)

(c) (d)

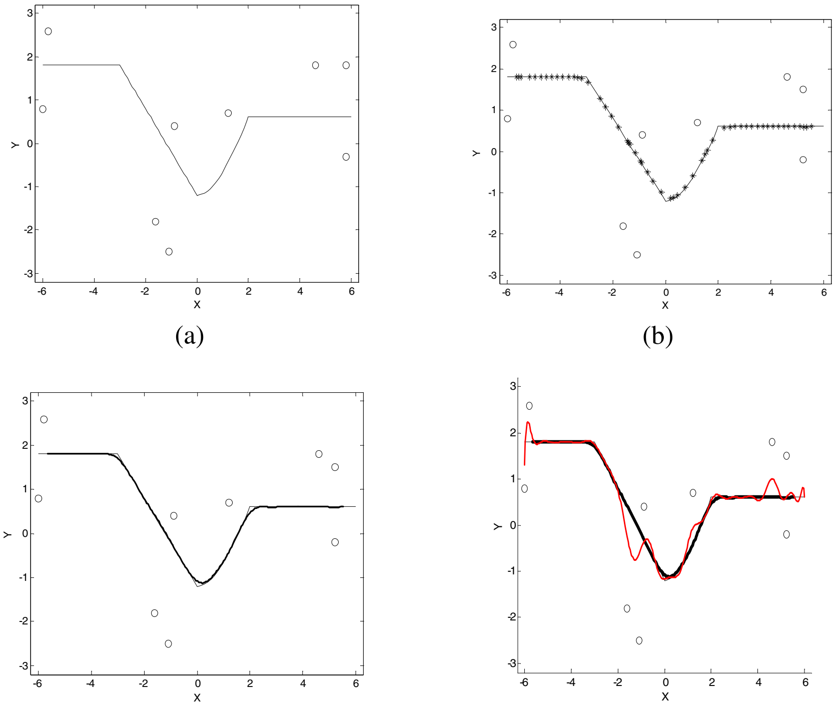
Fig. 3. Equation (20) with noise N (0, 0.4)

Within Fig. 3(a), there are 251 sampling data generated from equation (20) with Gaussian noise N (0,0.4). The circles represent sampling data with noise and the solid line represents the equation (20). Fig. 3(b) shows the result after running RFCM to generate cluster centers. In Fig. 3(b) stars denote the resultant cluster centers obtained by the RFCM. Fig. 3(c) shows the finial result obtained by the SVR using resultant cluster centers generated by RFCM. Fig. 3(d) shows the results from the proposed method in comparison with standard SVM using original sampling data. Within Fig. 3(d), the solid line denoted the equation (20), the solid dot line denoted the result obtained by the proposed method, and the dashed line denoted the result obtained by SVR using original sampling data. Fig. 3(d) shows the result obtained by the proposed method is closer the original function than the result generated by SVR using original sampling data. In addition, the totally execution time of the proposed method is 4.8 seconds and the SVR using original sampling data is 1192.9 seconds which is almost 248 times of the proposed method.

Example 2. A function used in [17] is defined as

$$y = f(x) = \begin{cases} 1.8 & x < -3 \\ -x - 1.2 & -3 \le x < 0 \\ 3e^{-0.1x}\sin(0.2x^2) - 1.2 & 0 \le x < 2 \\ 0.6 & 2 \le x \end{cases}, \qquad (21)$$

and x∈[-6,6]. There are 121 sampling data points generated from equation (21), and 9 artificial outliers are added in this function. Fig. 4(a) shows the function with outliers denoted by circles. Fig. 4(b) shows the result of the RFCM. Within Fig. 4(b), the stars denote the centers obtained by RFCM. Fig. 4(c) shows the finial result obtained by the SVR using resultant cluster centers generated by RFCM. Fig. 4(d) shows the results from the proposed method in comparison with standard SVM using original sampling data. Within Fig. 4(d), the solid line denoted the equation (21), the solid dot line denoted the result obtained by the proposed method, and the dashed line denoted the result obtained by SVR using original sampling data. Fig. 4(d) shows the result obtained by the proposed method is closer the original function than the result generated by SVR using original sampling data.



(a)

(b)

(c)

(d)

Fig. 4. The equation (21) with nine outliers

6 Conclusion

In this paper, a robust fuzzy clustering method is proposed to greatly mitigate the influence of noise and outliers in sampling data and then SVR method is used to construct the system models. The core idea in the proposed method is to use distance analysis between data points for searching the clustering centers, and the RFCM algorithm is proposed to greatly mitigate the influence of data noise and outliers. Finally, The resultant clusters of the RFCM are then used for SVR training. Three experiments are illustrated and their results have shown that the proposed approach has better performance and less execution time than original SVR method in various kinds of data domains with data noise and outliers.

References

1. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)
2. Gunn, S.R.: Support Vector Machines for Classification and Regression. Technical Report of University of Southampton (1998)
3. Burges, C.J.C.: A tutorial on Support Vector Machines for pattern recognition. Data Mining and Knowledge Discovery 2(2), 121–167 (1998)
4. Jen, J.T., Chuang, C.C., Su, S.F.: Support vector interval regression networks for interval regression analysis. Fuzzy Sets and Systems 138, 183–300 (2003)
5. Zhou, W., Zhang, L., Jiao, L.: Linear programming support vector machines. Pattern Recognition 35, 2927–2936 (2002)
6. Lijuan, C.: Support vector machines experts for time series forecasting. Neurocomputing 51, 321–339 (2003)
7. Gustavo, C.V., Luis, G.C., Javier, C.M., Jose David, M.G., Emilio, S.O., Luis, A.C., Jose, M.: Robust Support Vector Method for Hyperspectral Data Classification and Knowledge Discovery. IEEE Transactions on Geoscience and Remote Sensing 42(7), 1530–1542 (2004)
8. Hu, W.J., Song, Q.: An Accelerated Decomposition Algorithm for Robust Support Vector Machines. IEEE Transactions on Circuits and Systems—II: Express Briefs 51(5), 234–240 (2004)
9. Osuna, E., Freund, R.R., Girosi, F.F.: An improved training algorithm for support vector machines. In: IEEE Workshop Neural Networks for Signal Processing (NNSP 1997), pp. 276–285 (1997)
10. Lee, Y.J., Huang, S.U.: Reduced Support Vector Machines: A Statistical Theory. IEEE Transactions on Neural Networks 18(1), 1–13 (2007)
11. Lin, K.M., Lin, C.J.: A Study on Reduced Support Vector Machines. IEEE Transactions on Neural Networks 14(6), 1449–1457 (2003)
12. Wang, W., Xu, Z.: A heuristic training for support vector regression. Neurocomputing 61, 259–275 (2004)
13. Chung, C.C., Su, S.F., Hsiao, C.C.: The Annealing Robust Backpropagation(ARBP) Learning Algorithm. IEEE Transactions on Neural Networks 11(5), 1067–1077 (2000)
14. Jeng, J.T., Chuang, C.C.: Selection of Initial Structures with Support Vector Regression for Fuzzy Neural Networks. International Journal of Fuzzy Systems 6(2), 63–70 (2004)
15. Shieh, H.L., Yang, Y.K., Lee, C.N.: A Robust Approach of Fuzzy Clustering on Data Sets with Noise and Outliers. Cybernetics and Systems: An International Journal 37(7), 755–778 (2006)
16. Chuang, C.C., Jeng, J.T., Lin, P.T.: Annealing Robust Radial Basis Function Networks for Function Approximation with Outliers. Neurocomputing 56, 123–139 (2004)
17. Lee, C.C., Chung, P.C., Tsai, J.R., Chang, C.I.: Robust Radial Basis Function Neural Networks. IEEE Transactions Systems, Man, and Cybernetics Part B: Cybernetics 29(6), 674–685 (1999)

Performance Evaluation of Robot Motion Incorporating Uncertainties in Sensors and Motion

Dong Jin Seo[1], Nak Yong Ko[2], Gwang Jin Kim[2], Yongseon Moon[3], Youngchul Bae[4], and Seung-Woo Lim[5]

[1] Robotics Institute, Redone Technologies Co. LTD., Korea
commidi@gmail.com
[2] Department of Control and Instrumentation Eng., Chosun Univ., Korea
nyko@chosun.ac.kr
[3] School of Information Communication, Sunchon Nat'l Univ., Korea
moon@sunchon.ac.kr
[4] Division of Electrical Electronic Communication Chonnam Nat'l Univ., Korea
ycbae@chonnam.ac.kr
[5] Department of Information Communication., Dongkang College., Korea
lsw5802@naver.com

Abstract. This paper describes the performance evaluation of mobile robot motion. The evaluation is based on a probabilistic model of robot motion and sensor output. In the simulator, robot motion includes uncertainties in translational velocity and rotational velocity. Among various kinds of sensors, implemented in the simulator is a range sensor which is used widely for obstacle detection, map making, and localization. The sensor model includes measurement noise with Gaussian distribution around true range value, failure of detection, and unexplainable random measurement. The simulator is used to test a collision avoidance algorithm, map making, and motion coordination algorithm. The results from a simulated environment with uncertainty are compared with the results from a simulator with no uncertainty. In practical application of a mobile robot, major problems arise due to the unexpected uncertainty of motion and sensors. The simulator is useful to predict problems of robot motion before the robot is deployed in the field. For wider availability, the simulator can be further developed to include error models in various sensors.

Keywords: Performance evaluation, robot simulator, motion uncertainty, sensor uncertainty.

1 Introduction

A simulator which tests robot motion should take into account the uncertainty of robot motion and sensors. Though a robot works well in a simulated test, the robot can fail to work in real work environment. This is because the simulator doesn't consider the unexpected uncertainties in things like robot motion, the

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 271–280, 2009.

environment, detection errors, calculation errors and latency. As the complexity of robot hardware and software increases, it becomes harder to model the performance and robustness of the robot motion with certainty. So, a probabilistic method is required to consider the performance of an algorithm more practically.

Uncertainty is inevitable in robot systems. In the case of motion in a mobile robot, the motor driver cannot drive the motors exactly as commanded. The dimension of the wheel diameter, wheel base, and other mechanical elements also have uncertainties. Moreover the interaction of the wheel with the floor has uncertainties like slip. These uncertainties make the robot motion deviate from the expected trajectory. The sensor error also makes the motion more unpredictable[1]. Also, there is uncertainty in the robot's software. The robotic algorithm approximates the real system because of the lack of full information on the system and the computational limits. While many of the robotic approaches such as probabilistic robotics[1], evolutionary robotics[2], and behavior-based robotics[3] mainly take care of the uncertainties, simulators usually pay less attention to the uncertainties than the approaches.

This paper proposes a simulator which incorporates uncertainties in the sensor output and robot motion. The previous paper which implemented only motion uncertainty is not sufficient to test performance and predict possible problems of a robot in real application[4]. The uncertainty can cause serious problem in robot motion. This paper shows how an algorithm results in different motions in a given environment due to the uncertainty in sensor output as well as the uncertainty in robot motion.

The proposed simulator emulates a range sensor which is used widely in mobile robots for localization and obstacle detection. The range sensor output varies usually from the Gaussian noise around its true distance, failure of detection, and random noise spreading throughout the measurement range. These three kinds of errors are included in the simulator. Except for the error due to the unmodeled objects, the errors are modeled according to the book on probabilistic robotics[1].

There have been some simulators for robots: Webbots[5], MobotSim[6], Player/Stage[7], Gazebo[7], and Muros[8]. Webbots and Gazebo offer three dimensional environment. Though Webbots, Player/Stage, and Gazebo works in network environment, many of the simulators deal with only one robot rather than multiple robots. Though the safety problem becomes serious as the uncertainty of the robot and environment increases, they usually don't incorporate the uncertainty, and may overlook fatal problems. The simulator proposed in this paper is especially useful for verifying an algorithm for autonomous motion of a mobile robot in a harsher simulated environment.

Section 2 outlines the basic structure and features of the simulator, and Section 3 describes the sensor model which incorporates uncertainty in sensor output. The motion uncertainty is explained in Section 4. Section 5 describes some applications of the simulator for algorithm tests. Finally, Section 6 concludes the paper.

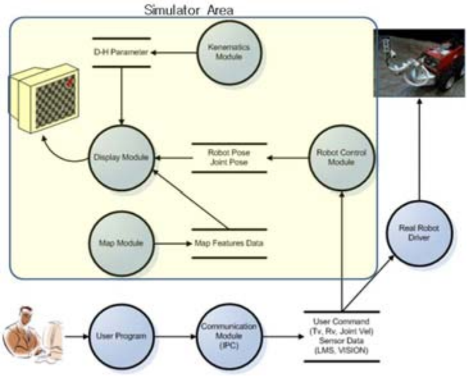


Fig. 1. Data flow diagram of the simulation program

2 Structure and Features of the Simulator

The simulator has a client-server structure based on TCP-IP. The robots are on the network which carries commands from the server to the robots and carries sensor data from the robots to the server. The simulator has virtual work environment with robots on the network, and the virtual robots communicate with the server. The virtual robots duplicate real robot systems, and the real robots can run in parallel with the virtual robots once they are logged onto the network.

2.1 Structure of the Simulator

Fig. 1 depicts the structure of the simulation program. The simulator is composed of the four functional units: robot control, kinematic design, driver for real robot operation, and communication unit. Their functions are as to the following.

Robot control: This deals with motor control, sensing, map management, and display of the robot and the environment. Uncertainties in robot motion are implemented in this unit.

Kinematic design: This generates robot poses and configuration in conjunction with the display module of the robot control unit.

Driver for real robot operation: This takes commands from the server and then operates the real robot.

Communication: This deals with communication between the robot control unit and the user program(server).

2.2 Features of the Simulator

The simulator has the following features which make the simulation more consistent with, and adaptable, in the real robot system.

Inclusion of uncertainty in sensor output and robot motion. The uncertainty is incorporated into the simulated sensor output and motion of the robot. The sensor output incorporates uncertainties of random noise and failure of sensing. The random noise consists of Gaussian noise and uniformly distributed noise. The pose of the mobile robot at a sampling time depends on the pose of the previous sampling time, the velocity command, and some noise effect. We assume that the noise has normal distribution with zero mean. The incorporation of uncertainties into the simulator plays a crucial role in evaluating and proving the robustness of an algorithm.

Versatility of the network: Since the simulator uses TCP-IP for communication, it can be used on a common ethernet network. It uses an IPC library for implementation of the communication[9].

Comprehensive robot control structure: The robot control module consists of the four functions: motor control, sensing, map management, and display. These four functions combine to implement the robot control functions.

Consistency between the simulator and the real robot system: The same command code does the same work on both the real robots and the virtual robots in the simulator. From the user's point of view, there should be no difference between the robots in the simulator and the real robots. That is to say, the software for robot control in the simulator should be able to control the real robot as well. To meet this requirement, the simulator is developed to have interface driver which drives the real robots as well as the robots in the simulator.

3 Sensor Model Implementing Uncertainty

The output of range sensors such as sonar and Laser range finder inevitably includes some error. Generally they measure the distance from the source to a reflecting object by evaluating the time of flight or phase shift of the signal. This method may give wrong measurement when the signal hits an edge or corner of an object. Also, the material of the object affects the measurement. Sometimes, they fail to detect an object. In case of sonars, cross talk and specular reflection make the measurement more challenging. The errors are inevitable in range sensing and this requires the sensor model to take care of uncertainties.

A sensor model which simulates a range sensor is proposed. The uncertainties proposed in the earlier literature include the following four types of errors: measurement noise, unexpected objects, failures, and random measurements[1]. The measurement noise arises from the limited resolution, atmospheric effects on the measurement signal, and other uncertainties. The unexpected objects account for unmodeled objects. The failures result in maximum-range measurements. Random measurements are the results of unexplainable measurements.

In our application three of the above mentioned four causes of error source are considered. One which is ignored is the *"unexpected objects."* One example of an unexpected objects is a person passing around the sensor. However, sensor output which reflects the presence of the passerby is not an error in itself. Therefore only the three remaining factors are included as error sources. The error model of these three are explained below. The details of the error model and adjustment of the model parameters are found in the literature by Thrun[1].

3.1 Error Due to Measurement Noise

The probabilistic error distribution due to measurement noise is modeled as the Gaussian with mean z_t^{k*}, and standard deviation σ_h. The z_t^{k*} is the true range value of the measurement. The measurement probability $p_h\left(z_t^k|x_t,m\right)$ which represents the probability that the measured range is z_t^k, is described as the following.

$$p_h\left(z_t^k|x_t,m\right) = \begin{cases} \eta\mathcal{N}\left(z_t^k; z_t^{k*}, \sigma_h^2\right) & \text{if } 0 \le z_t^k \le z_{max} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$$\mathcal{N}\left(z_t^k; z_t^{k*}, \sigma_h^2\right) = \frac{1}{\sqrt{2\pi\sigma_h^2}} e^{-\frac{1}{2}\frac{(z_t^k - z_t^{k*})^2}{\sigma_h^2}} \tag{2}$$

In these equations m means the environment map and x_t is the state or pose of the robot at time t. η is the normalizer which satisfies the following.

$$\eta = \left(\int_0^{z_{max}} \mathcal{N}\left(z_t^k; z_t^{k*}, \sigma_h^2\right) dz_t^k\right)^{-1} \tag{3}$$

3.2 Error Due to Failure

If a range sensor fails to detect an object, then the sensor returns its maximum allowable value z_{max}. This happens due to specular reflection in case of a sonar. In case of a Laser range sensor, it happens when the object to be sensed is black, or absorbs light, or is under bright sunlight. The probability distribution due to sensor failure is represented as the following.

$$p_m\left(z_t^k|x_t,m\right) = I\left(z = z_{max}\right) = \begin{cases} 1 \text{ if } z = z_{max} \\ 0 \text{ otherwise} \end{cases} \tag{4}$$

3.3 Error Due to Random Measurement

The sensors sometimes yield random values which are unexplainable. This happens due to cross-talk or bounce off in case of sonars. It is simply modeled as a uniform distribution throughout the entire measuring range of the sensor. The probability is described as the following.

$$p_r\left(z_t^k|x_t,m\right) = \begin{cases} \frac{1}{z_{max}} & \text{if } 0 \le z_t^k \le z_{max} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Combining the three probability distributions listed above, the probability distribution is found to be

$$p\left(z_t^k|x_t,m\right) = \omega_h \cdot p_h\left(z_t^k|x_t,m\right) + \omega_m \cdot p_m\left(z_t^k|x_t,m\right) \\ +\omega_r \cdot p_r\left(z_t^k|x_t,m\right) \tag{6}$$

where, ω_h, ω_m, and ω_r are the weightings satisfying $\omega_h + \omega_m + \omega_r = 1$.

4 Motion Model Implementing Uncertainty

In the motion model, the uncertainty of the robot motion is included. If we assume that there is no uncertainty in the motion of a mobile robot, the pose at time $t = t_{i+1}$, $(x_{i+1}, y_{i+1}, \theta_{i+1})^T$ will be calculated according to Equation (7).

$$\begin{aligned} x_{i+1} &= x_i - \frac{tv}{rv}\sin\theta_i + \frac{tv}{rv}\sin(\theta_i + rv \cdot \Delta t) \\ y_{i+1} &= y_i + \frac{tv}{rv}\cos\theta_i - \frac{tv}{rv}\cos(\theta_i + rv \cdot \Delta t) \\ \theta_{i+1} &= \theta_i + rv \cdot \Delta t \end{aligned} \tag{7}$$

where, $(x_i, y_i, \theta_i)^T$ is the pose of the robot at the i-th sampling time $t = t_i$, tv is the translational velocity command, rv is the rotational velocity command, and Δt is the sampling period. Since robots in practice move with uncertainty, Equation (7) is not adequate for use in the simulation.

Instead of Equation (7) which doesn't take the motion uncertainty into account, we use Equation (8) to incorporate the motion uncertainty into the simulator.

$$\begin{aligned} x_{i+1} &= x_i - \frac{\widehat{tv}}{\widehat{rv}}\sin\theta_i + \frac{\widehat{tv}}{\widehat{rv}}\sin(\theta_i + \widehat{rv} \cdot \Delta t) \\ y_{i+1} &= y_i + \frac{\widehat{tv}}{\widehat{rv}}\cos\theta_i - \frac{\widehat{tv}}{\widehat{rv}}\cos(\theta_i + \widehat{rv} \cdot \Delta t) \\ \theta_{i+1} &= \theta_i + \widehat{rv} \cdot \Delta t + \widehat{\gamma} \cdot \Delta t \end{aligned} \tag{8}$$

In Equation (8), $\widehat{tv}$, $\widehat{rv}$, and $\widehat{\gamma}$ are derived using the formulas in (9).

$$\begin{aligned} \widehat{tv} &= tv + \varepsilon_{\alpha_1 tv^2 + \alpha_2 rv^2} \\ \widehat{rv} &= rv + \varepsilon_{\alpha_3 tv^2 + \alpha_4 rv^2} \\ \widehat{\gamma} &= \varepsilon_{\alpha_5 tv^2 + \alpha_6 rv^2} \end{aligned} \tag{9}$$

In Equation (9), ε_{b^2} is a zero-mean error variable with variance b^2. The parameters $\alpha_1, \cdots, \alpha_6$ determine the probabilistic distribution of the uncertainty. They are robot-specific error parameters. They model the accuracy of the robot motion. The less accurate the robot motion is, the larger these parameters are[1].

Fig. 2 and Fig. 3 depict how the motion uncertainty affects robot motion in the simulator. A grid in the figures is a square with 0.5 meter long sides. The robot starts at the location numbered "1" where the robot is heading toward the

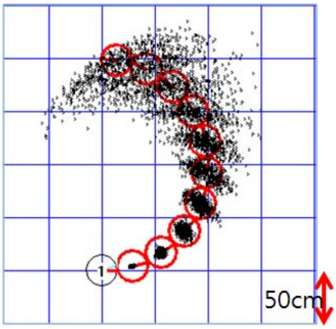


Fig. 2. Uncertainty in robot motion where $\alpha_1 = \alpha_2 = 0.1,\ \alpha_3 = \alpha_4 = 1,\ \alpha_5 = \alpha_6 = 0.01$

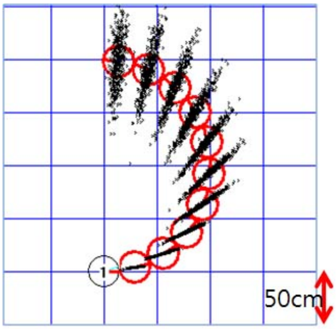


Fig. 3. Uncertainty in robot motion where $\alpha_1 = \alpha_2 = 1,\ \alpha_3 = \alpha_4 = 0.01,\ \alpha_5 = \alpha_6 = 0.01$

right. In each figure, the particles represent possible robot locations at 3 seconds after the start of motion. Rotational velocity corresponding to the first group of the locations from the start is $rv = 0.1rad/sec$. The rotational velocities are incremented by the $\delta rv = 0.1rad/sec$. We set the translation velocities to keep the radius of the robot path to 1 meter.

Fig. 2 shows position distributions with the motion uncertainty parameters $\alpha_1 = \alpha_2 = 0.1,\ \alpha_3 = \alpha_4 = 1,\ \alpha_5 = \alpha_6 = 0.01$ Fig. 3 shows those with the motion uncertainty parameters $\alpha_1 = \alpha_2 = 1,\ \alpha_3 = \alpha_4 = 0.01,\ \alpha_5 = \alpha_6 = 0.01$.

As the values of the parameters $\alpha_1, \cdots, \alpha_6$ change, the distribution of the pose changes. A comparison of Fig. 2 and Fig. 3 reveals how the parameters $\alpha_1, \cdots, \alpha_6$ affect the distribution of the robot location. In Equation (9), α_1 and α_2 determine the spread of the distribution in the translational motion. So, as α_1 and α_2 increase from 0.1 to 1, the distribution of possible locations spreads more along the radial direction. α_3 and α_4 affects the distribution in the rotational motion. Therefore, as α_3 and α_4 decrease from 1 to 0.01, the spread of possible locations decreases in circumference.

5 Applications of the Simulator

5.1 Obstacle Avoidance of a Mobile Robot

An obstacle avoidance algorithm based on the artificial potential field[10] is tested using the simulator. The collision-free path is compared with that generated by the simulator with no uncertainty. As will be shown, more of the property can be revealed through the simulation with uncertainty.

The artificial potential field method is used for a mobile robot to pass a corridor. The parameters used to incorporate uncertainties are listed in Table 1. Fig. 4 shows two trajectories each. The trajectory marked number "1" is the result from the simulation with uncertainty which is described as *Case 1* in Table 1. The trajectory numbered "2" is the result from the simulation without sensor uncertainty which is described as *Case 2*.

As can be seen obviously, when there is no uncertainty in the sensor model, robot motion is more predictable even if there is any uncertainty in robot motion. Uncertainty in the sensor model makes the trajectory more unpredictable.

5.2 Map Making

Fig. 5 shows the environment for map building using a range scan sensor. Fig. 6(a) shows the simulated sensor readings in the ideal case. In this case,

Table 1. Parameters for uncertainty

Cases	$\alpha_1 = \alpha_2$	$\alpha_3 = \alpha_4$	$\alpha_5 = \alpha_6$	ω_h	ω_m	ω_r
Case 1	0.01	0.1	0.001	0	0	0
Case 2	0.01	0.1	0.001	0.6	0.2	0.2

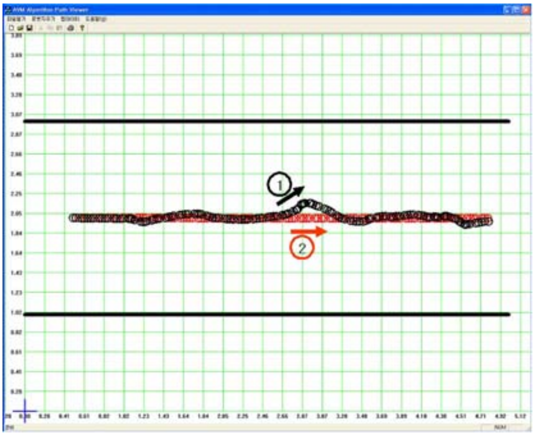


Fig. 4. The robot goes through a corridor. As shown by the trajectory marked by "1," the uncertainty in sensor model makes the trajectory unpredictable.

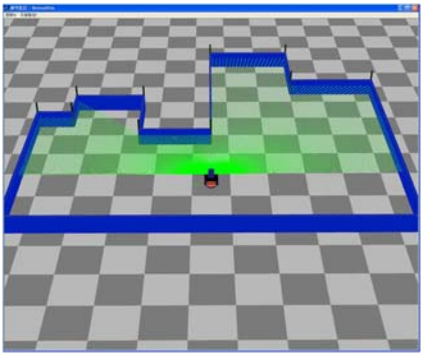
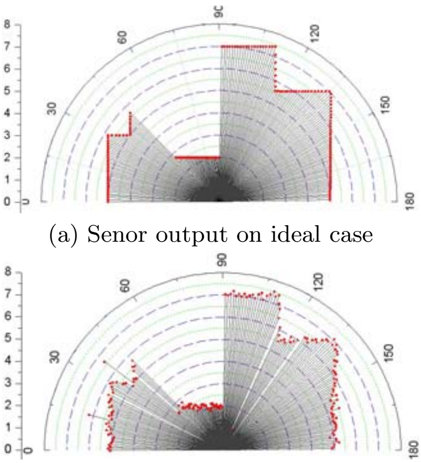


Fig. 5. Environment for sensor reading



(a) Senor output on ideal case

(b) Another senor output by the simulator

Fig. 6. Two simulated sensor readings at different cases

sensor data is exactly the same as the ray-cast data from the walls. As can be seen from Fig. 6(b), errors around the actual distance and random errors make the sensor output more practical to use in the test of an algorithm. Also, sensor readings change from time to time as is true for real sensors.

6 Conclusions

This paper describes a simulator which deals with robot motion and sensor output in a probabilistic approach. The results show that inclusion of uncertainties in a simulator enhances the applicability of the simulator in a real environment. The simulator has the following features.

1. With the simulator, robustness of robot motion can be verified more practically. Especially, it is possible to predict critical safety problems due to unstable interactions of the robot and the environment with sensing errors.
2. The simulator takes the motion uncertainty into account. So, it exhibits more practical performance compared with simulators which ignore the uncertainties.
3. The sensor model provides more realistic readings corrupted with noise which changes from time to time as is true for real sensors.
4. The simulator can be used to test the performance of localization algorithms, map making algorithms, and SLAM algorithms because it includes sensor uncertainty and motion uncertainty.
5. The simulator is useful in the ubiquitous environment where robots are in a network. Application in ubiquitous environments are made possible by the client-server structure which uses IPC. The IPC is based on TCP-IP which is prevalent in the worldwide internet environment.
6. It is not meant to simulate the robot motion as precisely as possible. The simulation is useful in analysing the performance of an algorithm probabilistically.

The uncertainties included in the paper have been well defined previously. However, the uncertainties are not compared with the real uncertainties. For further development, it is necessary to analyse the characteristics of actual uncertainties, and improve the models of the uncertainties. Also, inclusion of different kinds of sensors with their uncertainties is needed.

References

1. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press, Cambridge (2005)
2. Nolfi, S., Floreano, D.: Evlolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines. MIT Press, Cambridge (2000)
3. Arkin, R.C.: Behavior-Based Robotics. MIT Press, Cambridge (1998)
4. Ko, N.Y., Seo, D.J., Kim, G.J., Moon, Y.S., Bae, Y.C., Lee, S.M.: Simulator Implementing Uncertainties in Motion for Robots in a Network. In: Proc. IEEE International Conference on Mechatronics and Automation, China, pp. 2836–2841 (2007)
5. Cyberbotics Ltd., http://www.cyberbotics.com
6. MobotSoft Ltd., http://www.mobotsoft.com
7. Player/Stage Open Source Project, http://playerstage.sourceforge.net
8. Muros Multi Robot Simulator, http://www.cis.upenn.edu/~chaimo/MuRoS.htm
9. Simmons, R., Mitchell, T.: A Task Control Architecture for Autonomous Robots. In: Proc. Third Annual Workshop on Space Operations Automation and Robotics (SOAR 1989), USA, pp. 409–414 (1989)
10. Khatib, O.: Real-time Obstacle Avoidance for Manipulators and Mobile Robots. The Int. Journal of Robotics Research 5(1), 90–98 (1986)

Measuring Naturalness during Close Encounters Using Physiological Signal Processing

Yasser Mohammad and Toyoaki Nishida

Nishida-Sumi Laboratory, Department of Intelligence Science and Technology,
Graduate School of Informatics, Kyoto University, Japan
yasser@ii.ist.i.kyoto-u.ac.jp, nishida@i.kyoto-u.ac.jp

Abstract. Many researchers in the HRI and ECA domains try to build robots and agents that exhibit human-like behavior in real-world close encounter situations. One major requirement for comparing such robots and agents is to have an objective quantitative metric for measuring naturalness in various kinds of interactions. Some researchers have already suggested techniques for measuring stress level, awareness etc using physiological signals like GSR and BVP. One problem of available techniques is that they are only tested with extreme situations and cannot according to the analysis provided in this paper distinguish the response of human subjects in natural interaction situations. One other problem of the available techniques is that most of them require calibration and some times ad-hoc adjustment for every subject. This paper explores the usefulness of various kinds of physiological signals and statistics in distinguishing natural and unnatural partner behavior in a close encounter situation. The paper also explores the usefulness of these statistics in various time slots of the interaction. Based on this analysis a regressor was designed to measure naturalness in close encounter situations and was evaluated using human-human and human-robot interactions and shown to achieve 92.5% accuracy in distinguishing natural and unnatural situations.

1 Introduction

Personal robots that co-exist with untrained human users and partners are expected to increase dramatically in the near future. The acceptance of these robots in the human society depends on how these untrained human partners with whom they are to interact are perceiving them. For this reason it is important to have subjective evaluations of new interactive robot designs. Nevertheless, subjective evaluations are not completely satisfactory for comparing interactive robots for two reasons: First, even though subjective evaluations through questionnaires can cover to some extent the response of the subjects to the robot, it is very hard to compare results of such subjective evaluations coming from different research groups due to the difficulties in controlling environmental conditions across experiments. Second, the results of questionnaire are known to be cognition mediated and disagreement between reported and measured behavior is evident from many psychological studies [1]. In a subject with strong

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 281–290, 2009.

novelty effects, and sometimes high cognitive load like interacting with robots such limitations of subjective evaluations based on questionnaires become even stronger.

To avoid these problems objective metrics are needed to compare different interactive robot designs. Three main types of metrics can be used: psychological metrics based on psychological tests, behavioral metrics based on comparing the external behavior of the subjects to some predefined standard, and physiological metrics based on measuring the internal state of the subjects using their physiological responses. We focus on physiological metrics in this paper. Nevertheless, psychological and behavioral metrics are to be included in future research.

There are many possible dimensions of comparison between interactive robots, their appearance, their behavior, the partner's response to them, etc. One obvious measure of interactive robots' behavior is how human-like they are in terms of behavior and appearance, but it is not always the case that human-like behavior and appearance are considered as more *natural* by the partner [2].

As the goal of our metric is to compare interactive robots *as perceived by their human partners*, we focus on the partner's response to the robot.

Definition 1: *Natural* behavior of an agent is defined in this paper as the behavior that minimizes the elevation in stress level, cognition load, frustration, and anxiety of the partner during the interaction while maximizing his/her engagement.

A robot or agent that has more *natural* behavior according to this definition is more desirable than a robot that has less *natural* behavior in most normal situations. Using this definition it is possible to measure naturalness of agent's or robot's behavior by measuring the physiological response of its human partner to this behavior.

Even though many physiological signals (e.g. Galvanic Skin Response (GSR), Blood Volume Pulse (BVP), Respiration Rate (RR), Skin Temperature (ST), etc) were shown to correlate of different aspects of human internal states (e.g. stress, cognitive load, anxiety, frustration, engagement, etc) [3], [4], [5], [6], [7], [8]), in most cases the analysis was done using extreme controlled conditions in which differences in the internal state is intense enough to be captured by simple statistics of the physiological signal under processing. For example [4] and [5] used a 3D computer game environment, while [3] used a complex multimodal user interface with 12 different tasks of varying complexity. It is not clear that the correlations found in these extreme conditions can be reliably found in natural interaction situations. One contribution of this paper is to show that it does not and more complex signal processing and machine learning techniques are needed in order to uncover the differences in physiological response in natural interaction situations.

A second problem of available methods is that they put very little emphasize on the effect of the interaction context on the measured physiological signals because most of them are used to measure the response to an inanimate object (e.g. a computer interface). For example human-human interaction research show that normal interactions between humans go through different phases [9] including opening and closing phases. It is expected that the physiological response of every partner to the behavior of other partners will depend on the interaction phase in which this behavior takes place.

A third problem of physiological sensing of internal state is that in most cases different individuals have different responses to the same stimuli and these individual differences can be much higher than the differences that depend on the stimuli itself. For example [8] have shown that only 74% of the tested subject have statistically significant correlation between GSR level and arousal despite the wide usage of this signal to measure arousal [6].

In this paper we cast the problem of measuring naturalness of an agent's behavior, as a problem of measuring human's physiological response to that agent, and we cast this later problem as a regression problem in which a set of input features (calculated from the physiological signals) are fed to the system and the result is a floating point number ranging from zero to one measuring how *natural* was the behavior from the viewpoint of the subject. The result is a novel metric for behavior naturalness (as defined earlier in this section) that utilizes signal processing, data mining and machine learning techniques to alleviate the three aforementioned limitations of available physiological metrics. The proposed metric was able to distinguish natural and un-natural listening behavior in an explanation scenario with 92.5% accuracy.

2 Experimental Scenario

In this paper we are interested in evaluating the subject's response to the behavior of its partner during a natural close encounter situation. There can be many varieties of such situations and in this work we focus in the explanation scenario in which the subject is explaining the assembly/disassembly of some machine or device to a listener using verbal and nonverbal modalities. The listener can either be another human subject or a robot.

This explanation scenario was selected because of its importance for HRI applications (i.e. learning by demonstration, knowledge media robots [2], companion robots etc). To reduce the number of participants needed to get robust results we used only diadic interactions. To make the interaction context as normal as possible only noninvasive sensors were used, and the instructor was familiarized with the task before the experiment. The time of the sessions was not limited by the experimenter to reduce the effect of stress due to time constraints on the results.

Some researchers in HCI and HRI community studied various physiological measures of internal state of human subjects. In [4], an evidence that physiological data correlate with task performance data in a video game was found: with a decrease of the task performance level, the normalized galvanic skin response (GSR) increases. In addition, physiological data were mirrored in subjective reports assessing stress level. [3] have also shown that GSR is positively correlated with increased cognitive load when comparing subjects response to a multimodal and a unimodal user interface. Skin conductance is also correlated with affective arousal [6], frustration [5], and engagement [7]. In [8] 74% of the subjects exhibited this correlation. Two channels related to the skin conductance were used research: Galvanic Skin Response [GSR] and Skin Conductance Level [SCL].

The heart rate has been used to differentiate between positive and negative emotions [10]. Heart rate variability is used extensively in human factors literature as an indication of mental effort and stress in high stress environments [11]. In this research we use Blood Volume Pulse (BVP) from which both heart rate and heart rate variability can be inferred.

Respiration is most accurately measured by gas exchange in the lungs but the sensors used to measure it accurately prevents moving and talking. Instead a stretch sensor can measure chest cavity expansion associated with breathing. This kind of sensors is noninvasive and was used in this experiment. Respiration is believed to be too slow for reflecting real time change in the internal state of humans but in [1] we showed that using appropriate processing it can be a reliable physiological differentiator between the response of an instructor to an attentive and inattentive listener. In this paper we use a stretch sensor to measure respiration response of the subjects at real time.

In summary we used three physiological sensors(skin-conductance, BVP, and Respiration) that are believed, based on related research, to have high correlations with the dimensions of response to *natural* behavior as defined in this paper (Definition 1). The sampling rate used was 100Hz.

3 Data Collection and Preliminary Analysis

The dataset consists of 66 sessions involving 22 human subjects acting as instructors explaining the assembly and disassembly of one of two objects. Every subject did three consecutive sessions with at least five minutes of rest between sessions. The three listeners for every subjects were: a human subject who was asked to attend to the explanation and be natural in his/her nonverbal behavior, a human subject who was asked not to attend to the explanation and to use unnatural nonverbal behavior while listening, and a humanoid Robovie II robot that was only moving its eyes randomly during the interaction. The three physiological signals studied in this work were measured before, after, as well as during every session using noninvasive Polymate physiological signal acquisition sensors. Both human listeners were wearing cold-masks to reduce the effects of facial expression and appearance on the results and to make human and robot listening sessions more comparable. The cold-masks used are very common in Japanese society not to invoke any special response from the subjects. The order of the sessions was shuffled to remove any ordering effects.

Table 1 shows the results of t-test analysis of the mean of GSR (usually used to measure stress level, cognitive load, and arousal), mean HRV (used to measure mental effort and stress), and Respiration Rate in the three conditions of this experiment. As clear from the table there is no statistically significant difference between the subject's response to natural, unnatural, and robot behaviors that can be deduced from these features. This negative result show that traditional

Table 1. Difference in means of GSR, HRV and Respiration Rate between different conditions in the experiment. For every two conditions the t and p values are shown.

	Mean GSR	Mean HRV	Mean RR
Natural vs. Unnatural sessions	t=1.833, p=0.074	t=1.25, p=0.192	t=0.228, p=0.192
Natural vs. Robot sessions	t=1.391, p=0.171	t=0.532, p=0.597	t=0.770, p=0.445
Unnatural vs. robot sessions	t=0.952, p=0.347	t=0.849, p=0.401	t=0.577, p=0.567

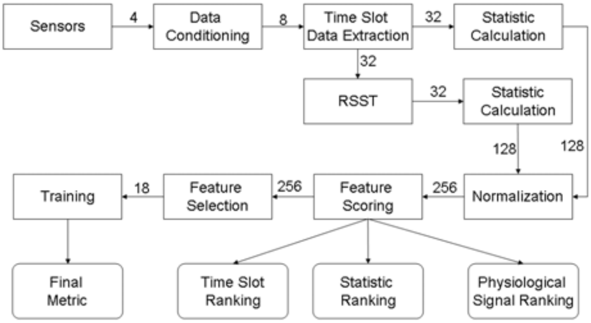


Fig. 1. Data processing steps applied to the physiological signals collected during the 66 sessions. Each line shows the dimensionality of the data transferred between processing blocks.

stress and cognitive load physiological measures cannot be used directly in determining the subject's response to partner's behavior in natural close encounter settings like the explanation situation used in this experiment. These results agree with what was reported in [12] in other interaction contexts. In the rest of this paper we provide a solution of this problem by fusing features of multiple signals in appropriate times of the interaction using a neural network.

4 Data Processing

The goals of this work was three fold: Firstly, to assess the usability of various physiological signals in evaluating *naturalness* of behavior as defined in this work (Definition 1). Secondly, to support or reject the hypothesis that different time slots of the interaction may not be of the same importance in this evaluation. Thirdly, to use the most important signals and time slots to drive an objective metric of *naturalness.*

Fig. 1 shows the processing steps applied to signals collected during this experiment. As physiological sensors are usually very sensitive, outlier removal is a must when dealing with them. To remove outliers we used Rosner's many-outliers test [13] applied to the logarithm of the input data (as the original data was not normally distributed). After the outliers are removed the signals were smoothed using a Savitzky-Golay filter of second degree. The following signals were then calculated from the smoothed sensor data: Heart Rate (HR), Heart

Rate Variability (HRV) as well as raw pulse data (P) were calculated from BVP data. Skin Conductance Level (SCL) and Galvanic Skin Response (GSR) were calculated from skin-conductance sensor data. Respiration Rate (RR), Respiration Rate Variability (RRV), and raw respiration pattern (R) were calculated from the respiration sensor. This leads to a total of eight physiological signals to be processed.

To find the effective interaction time slots 32 time series were calculated from the eight physiological signals of every session by using the first two minutes, the middle four minutes, the last three minutes, and the total interaction time. This two minutes boundary was decided based on our previous research in adaptation of human behavior to nonhumanoid robots [14] which show that the adaptation started after roughly two minutes of the interaction time.

For every one of these 32 time series the following features were extracted: mean (MEA), standard deviation (STD), minimum (MIN) and maximum (MAX). leading to a 128 features for every session. These are the statistics usually extracted from physiological data when measuring internal state [3], [6], [15].

The preliminary analysis presented in the preceding section showed that under normal interaction conditions, the mean of physiological signals did not provide much information about the internal state of the subject. These results were shown to be true with the same extent to the minimum and maximum of all of the signals measured. On the other hand standard deviation showed little improvement as it was less for natural condition but the difference was not statistically significant. Based on this point and the fact that negative emotions cause irregularities in respiration pattern [16], we decided to estimate the change in the underlying generating process of every physiological signal at every point of time and to use this as an extra source of information about the change in the internal state of the subject.

To measure changes in the underlying dynamics of every signal the Robust Singular Spectrum Transform (RSST) [17] was applied to every time series and the number of local maxima per second (NLM), maximum number of local maxima per minute (MLM), mean (RMEA) and standard deviation (RSTD) of the response were calculated. This led to 128 more features for every session totaling 256 features per session. The RSST transform is a data mining algorithm for unsupervised change-point detection in single dimensional time-series. Given a time series $x\left(t\right)$ of length T, the RSST transform of this time series is another time series $\left(x_{rsst}\left(t\right)\right)$ of the same length $\left(T\right)$ where $0 \leq x_{rsst}\left(t\right) \leq 1$ for $t = 1 : T$. Higher response means higher probability that the underlying generation process of the time series is changing. This transform have been shown to provide an effective measure of change when applied to synthetic data as well as respiration data in [17]. Due to lack of space, details of RSST will not be given in this paper.

Previous research in HCI suggests that individual differences between subjects can partially be eliminated by normalizing the statistics driven from physiological signals during the experiment by the same statistics from the same signals before the interaction using:$S_N = \frac{S-S_b}{S_b}$, where S_N is the statistic/feature after normalization, S is the statistic/feature before normalization and S_b is the statistic/feature in the five minutes immediately preceding the session. By normalizing over the base statistic before every session rather than the base values obtained before the whole experiment as done in [5], effects of earlier sessions do not *leak* to the features calculated at later sessions.

After normalization we scored every one of the 256 normalized features according to their importance based on their effect on the principal components of the dataset as follows:

1. The data of all sessions are arranged into a 66×256 matrix where rows correspond to different sessions and columns correspond to various normalized features. This matrix is called D.
2. D is then factorized using singular value decomposition using $D = USV^T$, where S is a diagonal matrix with singular values in its diagonal and U, V are the right and left singular values respectively
3. The raw score of every feature is calculated as: $r_i = \sum\limits_j s_{jj} \times |v_{ij}|$ where v_ij is the element at row i and column j of the matrix V.
4. These raw scores are then normalized to values from 0 to 1 using:$R_i = \frac{r_i - \min(r_i)}{\max(r_i) - \min(r_i)}$, where R_i is the final score of every feature i

The main idea behind this procedure is that when a feature is more important in explaining the variance in the data its value will be multiplied with larger absolute factor in the singular vectors corresponding to the singular values with large values. The normalization step makes sure that having larger range does not give a feature higher score. Based on these feature scores, it is possible to assess the importance of every signal, time slot, and statistic in explaining the variance in the physiological data simply by summing the scores of all the features driven from them.

All the features with scores over 10^{-6} (18 features shown in Table 2) were then selected and used for training. As a final final normalization step zscore normalization was applied to selected features before training as follows: $F_N = \frac{S_N - \mu_N}{\sigma_N}$, where F_N are final features used for training, S_N are the normalized features, μ_N is the mean of feature S_N and σ_N^2 is its variance. The 44 sessions representing natural and unnatural behavior were divided into 12-session testing and 32-session training sets subject to the constraint that all sessions of the same subject are used either for testing or training. The 32 training sessions were further divided into four groups of 8 sessions (4 subjects) each. 4-fold cross validation was then used to train a feed forward neural network with 18 inputs, 9 hidden and one output neuron to regress a measure of naturalness based on the labels of the training set using backpropagation. The resulting neural network is then used to evaluate the naturalness level of the 12 sessions in the testing set and the 22 robot sessions. The results of this processing is provided in the following section.

5 Results and Discussion

Table 2 shows the only eighteen features that had scores over 10^{-6}. The cumulative score of features from skin conductance was 1.86 compared to 1.37 for

Table 2. Scores of the 18 effective features

Statistic	NLM	MLM	STD	RSTD	RSTD	NLM	STD	RSTD	MEA
signal	R	GSR	SCL	PR	PR	PRV	SCL	PR	GSR
Time Slot	2	2	2	2	0	2	0	0	0
Score	1	0.6087	0.5784	0.4607	0.2966	0.2727	0.261	0.2068	0.1853

Statistic	NLM	STD	MLM	MEA	MEA	MLM	MEA	MLM	MEA
Signal	R	RRV	SCL	HRV	RRV	GSR	RRV	GSR	HRV
Time Slot	0	2	2	2	2	0	1	1	0
Score	0.1316	0.0952	0.0899	0.0741	0.0736	0.0735	0.0704	0.0623	0.0615

both pulse and respiration. This shows that relying in any one physiological signal is not enough for distinguishing subjects response to natural and unnatural behavior.

The cumulative score of features measured from the RSST was 3.203 compared to 1.4 for features that were calculated directly from the physiological signals which means that RSST based features are more than 2.88 times more effective in measuring naturalness in close encounters. They may be partially attributed to the fact that the stress and cognitive loads in these situations are not high enough to affect the *levels* of the physiological signals but they can appear as fluctuations in the dynamics of these signals which can be detected using RSST.

The cumulative score of different time slots revealed another interesting feature. Of the eighteen effective features in the dataset, there were no features measured based solely on the third slot of the session time, while the cumulative score of features measured in the first two minutes of the interaction was 0.1327 compared to 3.2533 for the middle four minutes of the interaction. This shows that the middle four minutes of the interactions in this corpus are 32 times more effective in measuring naturalness compared to the first two minutes of the interaction. This may partially attributed to the alignment period in the beginning of the interaction [14] during which the partners may be still searching for the adequate interaction protocol to be used in this session. After this alignment period the partners may become more *sensitive* to unnatural behavior of each other as a specific interaction protocol is now in effect. By the end of the interaction the each partner already expects either natural or unnatural behavior from the other and this may be the reason that the last two minutes of the interaction did not contribute to effective differentiation between the two types of sessions.

Fig. 2 shows the distribution of the response of the trained neural network to the testing dataset including 6 natural interaction sessions, 6 unnatural interaction sessions and the 22 human-robot interaction sessions. The average response to natural sessions is 0.733 (std. dev. 0.145) compared to 0.279 (std. dev. 0.212) for unnatural sessions and 0.387 (std. dev. 0.095) for robot sessions. Applying Factorial t-test showed that these differences between natural and robot sessions are statistically significant. Comparing natural and unnatural sessions: t=4.322, p=0.002. Comparing natural and robot sessions: t=5.530, p=0.001. Comparing unnatural and robot sessions: t=1.844, p=0.077. These results show that the

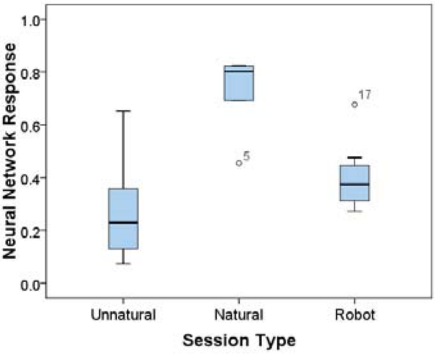


Fig. 2. The response of the trained neural network to the testing dataset including 6 natural interaction sessions, 6 unnatural interaction sessions and the 22 human-robot interaction sessions

proposed system was able to accurately distinguish natural and unnatural sessions and assigned the robot that randomly moves its eyes to unnatural behavior. By using a threshold value of 0.66 the accuracy of the trained network in classifying natural and unnatural sessions becomes 92.5% corresponds to a single mistake in the 12 testing sessions.

6 Conclusions

This paper explored the usefulness of various kinds of physiological signals and statistics in distinguishing natural and unnatural partner behavior in a close encounter situation and found that combining skin conductance, pulse, and respiration based physiological signals can be useful for measuring naturalness in these situations while each of them alone did not provide an effective measure. The paper also explored the usefulness of various statistics in various time slots of the interaction and found that change detection driven statistics using the RSST algorithm were more effective than the statistics driven directly from the signals while the middle four minutes of the interactions were more effective than the beginning and closing two minutes of the interaction in measuring naturalness. Based on this analysis a regressor was designed to measure naturalness in close encounter situations and was evaluated using human-human situations and shown to achieve 92.5% accuracy. Applying the same system to the response of the subjects to a humanoid robot that uses random gaze during the interaction showed that this behavior is perceived as unnatural behavior.

In the future this technique will be applied to compare different designs to of a listener robot and will be compared with subjective and psychological evaluations. Another possible extension of the work presented in this paper is to use the proposed naturalness measure to adapt robot behavior in real-time.

References

1. Mohammad, Y., Xu, Y., Matsumura, K., Nishida, T.: The h^3r explanation corpus:human-human and base human-robot interaction dataset. In: The fourth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2008) (December 2008)
2. Qazi, Z., Wang, Z., Haq, I.: Human likeness of humanoid robots exploring the uncanny valley. In: International Conference on Emerging Technologies, ICET 2006, pp. 650–656 (2006)
3. Shi, Y., Choi, E.H.C., Ruiz, N., Chen, F., Taib, R.: Galvanic skin respons (gsr) as an index of cognitive load. In: CHI 2007, April 2007, pp. 2651–2656 (2007)
4. Mandryk, R.L., Inkpen, K.M.: Physiological indicators for the evaluation of co-located collaborative play. In: CSCW 2004 (November 2004)
5. Lin, T., Hu, W., Omata, M., Imamiya, A.: Do physiological data relate to traditional usability indexes? In: OZCHI 2005 (November 2005)
6. Lang, P.J.: The emotion probe: Studies of motivation and attention. American Psychologiest 50(5), 372–385 (1995)
7. Mower, E., Feil-Seifer, D.J., Mataric, M.J., Narayanan, S.: Investigating implicit cues for user state estimation in human-robot interaction using physiological measurements. In: 16th International Conference on Robot & Human Interactive Communication, August 2007, pp. 1125–1130 (2007)
8. Bradley, M., Lang, P.J.: Measuring Emotions: Behavior, Feeling and Physiology. In: Cognitive Neurosceince of Emotion. Oxford University Press, New York (2000)
9. Argyle, M.: Bodily Communication. New Ed edition, Routledge (2001)
10. Papillo, J.F., Shapiro, D.: The Cardiovascular System. In: Principles of Psychophysiology: Physical, Social, and Inferential Elements. Cambridge University Press, Cambridge (1990)
11. Rowe, D.W., Sibert, J., Irwin, D.: Heart rate variability: Indicator of user stateas an aid to human-computer interaction. In: Conference on Human Factors in Computing Systems (CHI 1998) (1998)
12. Kidd, C.D., Breazeal, C.: Human-robot interaction experiments: lessons learned. In: Robot Compaions: hard problems and open challenges in Robot Human Interaction symbosioum of Social Intelligence and Interaciton in Animals Robots and Agents, April 2005, pp. 141–142 (2005)
13. Rosner, B.: On the detection of many outliers. Technometrics 17(2), 221–227 (1975)
14. Mohammad, Y., Nishdia, T.: Human adaptation to a miniature robot: Precursors of mutual adaptation. In: IEEE 17th International Symposium on Robot and Human Interactive Communication (IEEE Ro-Man), pp. 124–129 (2008)
15. Liao, W., Zhang, W., Zhu, Z., Ji, O.: A decision theoretic model for stress recognition and user assisstance. In: AAAI 2005, pp. 529–534 (2005)
16. Stern, R.M., Ray, W.J., Quigley, K.S.: Psychological Recording, 2nd edn. Oxford University Press, Oxford (2001)
17. Mohammad, Y., Nishida, T.: Robust singular spectrum transform. In: IEA/AIE (accepted, 2009)

A Neural-Evolutionary Model for Case-Based Planning in Real Time Strategy Games

Ben Niu, Haibo Wang, Peter H.F. Ng*, and Simon C.K. Shiu

Department of Computing,
The Hong Kong Polytechnic University,
Hum Hom, Kowloon, Hong Kong, P.R. China
polyust_niuben@yahoo.com,
{cshbwang,cshfng,csckshiu}@comp.polyu.edu.hk

Abstract. Development of real time strategy game AI is a challenging and difficult task because of the real-time constraint and the large search space in finding the best strategy. In this paper, we propose a machine learning approach based on genetic algorithm and artificial neural network to develop a neural-evolutionary model for case-based planning in real time strategy (RTS) games. This model provides efficient, fair and natural game AI to tackle the RTS game problems. Experimental results are provided to support our idea. This model could be integrated with warbots in battlefields, either real or synthetic ones, in the future for mimic human like behaviors.

Keywords: Case-based planning, real time strategy (RTS) games, genetic algorithm, artificial neural network.

1 Introduction

Computer games provide an ideal platform for Artificial Intelligence (AI) research. For example, traditional techniques such as finite-state machines, A* searching, logic- and rule-based reasoning have been proved extremely useful for developing game AI applications. Real Time Strategy Games, such as Warcraft (Blizzard Entertainment, Inc.) and Age of Empires (Microsoft, Inc.), relying heavily on AI, have attracted millions of users world-wide playing online over the Internet. These games provide not only a platform for entertainments but also an interactive environment for making friends, chatting, advertising, trading and other social activities. In these RTS games, there are many attractive features, e.g., interesting 3D landscapes, attractive avatars, and virtual currencies and resources, which are produced, controlled, manipulated or collected by the game players and third parties. However, the current architecture of game applications doesn't support well the utilization of user contributed contents to get better game playability. The difficulty is that the AI algorithms now used in computer games are mostly heuristic-based. The portability of the algorithms is quite poor. For example, in the RTS game a team of the computer generated characters can perform well on a predefined map by following the heuristic rules specified by the

[*] Corresponding author.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 291–300, 2009.

game developer but may later fail to behave correctly on a new map created by the users due to the lack of the rules to handle this new situation. To address this deficiency, we suggest building a machine learning component into the game so that the characters are able to learn from data directly for decisions and actions without using the heuristics. We have built a prototype system to test our idea. The system is designed to mimic the human ability that people can learn and plan to solve problems by first trying and working hard exhaustively to find out the 'good' solutions and then remembering the correct answers to make decisions quickly in the future. The results demonstrated the feasibility and the effectiveness of our technique. This paper is organized in six main sections. Section two describes the problem and its background. Section three proposes our method and algorithm. Section four presents our experimental results, and the conclusion is given in section five.

2 Problem Background and Description

The current game AI approach is something like "God" AI, i.e., it knows all the user actions. But under this arrangement, players will lose interest quickly. For example, when the game logic is designed and represented using finite state machines [1], which is usually based on inputting the hard-coded parameters, players may lose interest quickly once they have discovered that repeated patterns are being used. Moreover, this process requires a lot of time to fine tune the parameters and often relies on trial-and-error. Once the players discovered that the game behavior is somewhat "hard-coded", they may be disappointed. Therefore, machine learning approach may be a better choice to introduce human like behavior. Hsieh [2] used 300 professional game players' replays to train the decision system in one single battlefield. The result is a much more human like robot behavior in the battle. Zanetti [3] and Nicholas Cole [4] used genetic algorithm for first person shooting game AI development. Salge [5] also applied machine learning technique on turn base strategic game. Louis [6] developed a Case-injected genetic algorithm for NPC action in RTS game. These are some of the researches that tell us AI is getting more popular in RTS games. We will extend this machine learning idea and present our techniques and simulation data in the following sections.

2.1 Genetic Algorithm

Genetic algorithm (GA) was first introduced by John Henry Holland [7] in 1975. It is a search technique to find an approximate solution based on the concept of human evolution. GA is designed for large, non-linear, and poorly-understood search spaces in which heuristic-based search is found to be unsuitable. In RTS games, GA can be used as a guided random walk technique based on natural selection to find the optimal solution. However, using GA to develop RTS game strategy also faces some difficulties. First is how to design the GA encoding scheme for large and complex problems such as computer games, i.e., it is difficult to represent the whole battlefield using binary DNA. Second is how to design the fitness function for evaluating the chromosomes. Third is the real time constraint.

2.2 Artificial Neural Network

Artificial neural network (ANN) is a mathematical model or computational model based on biological neural networks. ANN has been used as an adaptive learning system that changes its weights based on external information. Again, ANN has rarely been used in game development. Only recently, David M. Bourg and Glenn Seemann [8] have introduced some discussions to use ANN in games.

3 Problem Description and Proposed Approach

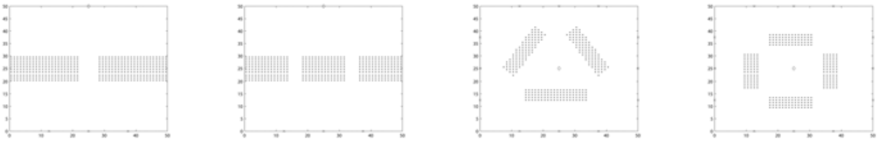
There are many possible game plays in any RTS games, in this paper, we have chosen a base defense scenario as the testing bed of our technique (e.g., Tower Defense in Warcraft). The situation is described as below.

1. Two teams are created in a battlefield. One is the attack team (enemy). Another one is the defense team (player).
2. Enemy must move to the base of the defense team and attack.
3. Defense team is able to set up a number of cannons (e.g., ten or fifteen) in the battlefield to kill the enemy when they are approaching their base.
4. The goal is creating the maximum casualty to the enemy no matter which path they choose to approach the base.

3.1 Divide and Conquer

Since it is difficult to design one binary chromosome in GA to represent the entire RTS game, our concept is dividing the game play into smaller parts. This also makes the fitness function easier to be formulated. Therefore, in this research work, the chromosome encoding focuses on base defense only.

Similar to the concept of spatial reasoning Frobus [9], battlefields can be also divided into different smaller subfields and with different canyons. In our simulation, we can classify a battlefield into four different smaller subfields (or maps) as shown in Fig. 1.



1 Canyon 2 Canyons 3 Canyons 4 Canyons

Fig. 1. Four different canyons. They represent the common landscape in RTS game.

To divide the battlefield into smaller subfields also gives us two advantages. First, the fitness function can work more effectively. Open area without canyons will not be considered as it will not affect the solution, but increase the run time because the search space increases.

3.2 Genetic Algorithm

GA requires two crucial components that need to be designed. First is the binary encoding scheme. It is the genetic representation of the solution.

The length required for the encoding depends on the area of the battlefield and the possible barriers. Here we use an example to demonstrate the encoding idea and is shown in Fig 2. A map with 5x5 units is used and 5 cannons to be set in the battlefield. Here, white circle represents the open area that a cannon can be set (i.e., located). The black circle represents the barrier that is unable to set up the cannon. The grey circle represents the position in which the cannons are set up. The total bits of the encoding depend on the size of the open area that is usable for setting up the cannons. (i.e., Encoding bit = Total area – Barrier area). In this example, it will be 13 bits. Five cannons are set on the map randomly as shown in Fig. 2. The encoding of each distribution of cannon is formulated as a chromosome.

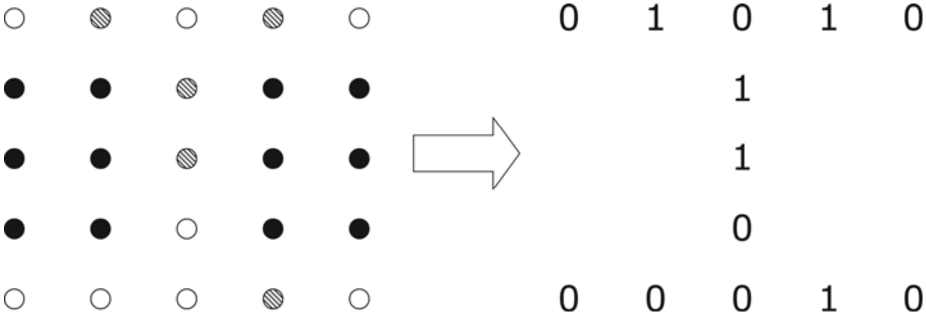


Fig. 2. Demonstration of the encoding in GA

In this simulation, 50x50 units map will be used and 15 cannons will be set on the map. The length of the chromosome is shown in Table 1.

Table 1. Length of chromosome

Canyon	Chromosome length (bits)
1	50 x 50 – 440 = 2060
2	50 x 50 – 420 = 2080
3	50 x 50 – 268 = 2232
4	50 x 50 – 280 = 2220

Another crucial component is a fitness function to evaluate the solution. In our simulation, 50 chromosomes are chosen in each generation for better and quicker convergence of the GA process. The evaluation function measures the enemy's casualty (or damage caused to them).

After the 15 cannons have been set up, the enemy will try to find the best path to go to attack the player's base. Enemy is given a certain velocity (v). It will receive damage within the attack range of the cannon. Damage (hit point) is calculated using equation (1) above. The total damage is calculated by summing up all individual damages caused by different cannons, and becomes the fitness value of this simulation. The higher the damage did to the enemies, the better the fitness of the cannons' positions. Cannon positions will be ranked by this fitness values and used to produce off-springs.

$$Damage = \frac{Travel\ distance\ (d)}{Velocity\ (v)}\ x\ Canon\ Power\ \text{(hit point per second)} \tag{1}$$

4 Testing Arrangements and Results

This simulation is done using an Intel Pentium IV 2.4GHz machine with 1.5 GB Ram under Windows XP. MathWorks Matlab 7.0 is used as the simulation tool. The result is shown in Fig. 3.

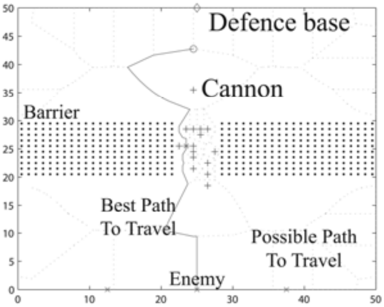


Fig. 3. Cannon distribution (Optimized by GA)

4.1 Results in Different Generations

The evaluation function is designed based on the damage done to the enemies. During the simulation, the fitness does not have a significant change after 100 generations as shown in Fig. 4, i.e., the result stabilized. This stabilization result is similar to Chuen-Tsai Sun [10] and Yi Jack [11] results.

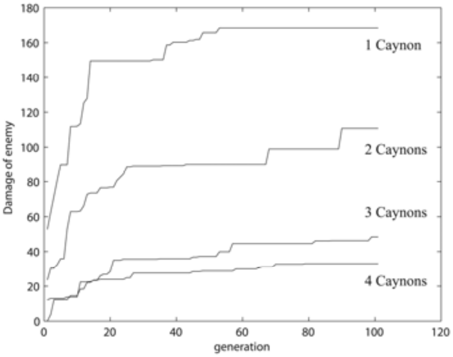


Fig. 4. Damage on enemy for different generations

4.2 Using ANN to Speed Up the Process

GA is a time consuming process. Table 2 is the run time of using GA with different number of canyons. The average run time of using GA is 1947 seconds which is unacceptable in RTS games and real world battlefields.

Table 2. Run time of using GA (Population size: 50, Generations: 150)

No. of Canyon	Run time of GA (second)
1	1249.5
2	1279.3
3	2606.9
4	2650.4

To overcome this problem, we suggest using artificial neural network (ANN) to speed up the whole process. Our proposed neural-evolutionary model is given in Fig. 5. First, the given terrain information for deciding of cannon distribution is encoded as a chromosome. Different generations of chromosomes are produced. Through the fitness evaluation the best off-springs (i.e., cannon distributions) will be produced. These best cannon distributions can be used as inputs to ANN for training. Afterwards, cannon location prediction can be suggested quickly and directly by ANN for new battlefields having different landscape.

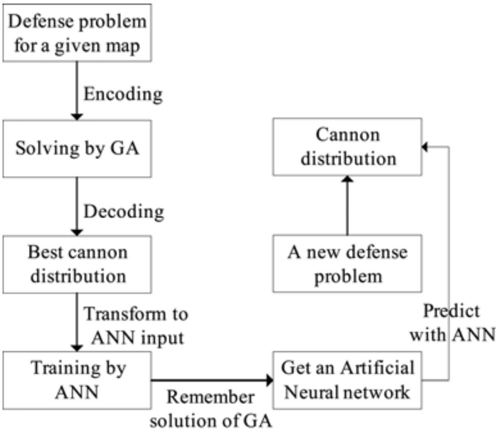


Fig. 5. Neural-Evolutionary Model

The idea of case-based planning in our model is as follows. We use the four best solutions (cases) that correspond to 1, 2, 3 and 4 canyons obtained from the GA training previously as the inputs to the ANN. For each case, every point with a certain radius will be treated as one of the inputs to the ANN. For example, Point A as shown in Fig. 6 will be one input. The 8 points around Point A will be encoded. “-1” represents the barrier while “1” represents the open area and the final digit “d” represents the distance between the Point A and the base of the player.

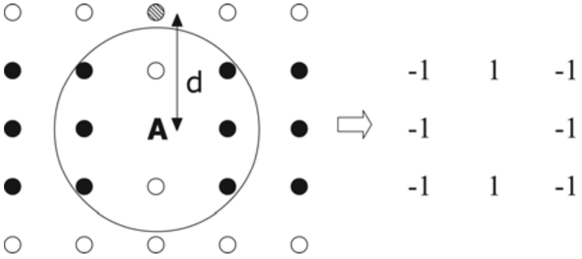


Fig. 6. Encoding of Point A

The encoding becomes [-1 1 -1 -1 -1 -1 1 -1 d]. The objective of the ANN learning is to approximate the best location function developed by us in (2). It is defined in such a way that the higher the value, the better the location for setting up the cannon. It is based on the relationship among the current landscape of Point A, the cannon locations previously found using GA and the distance to the base.

$$f(Point\ A) = \sum_{K=1}^{n} e^{-\frac{(A_1-K_1)^2+(A_2-K_2)^2}{r}} \tag{2}$$

(A_1, A_2) is the coordinate of point A. n is the total number of cannon, i.e., n=15 in this case. (K_1, K_2) is the coordinate of cannon K, and r is a parameter for controlling the spread of *f(Point A)*. The ANN training is using back-propagation and log-sigmoid output function. The number of the hidden layer is calculated as the square root of the encoding string’s length. In this example, it will be $\sqrt{9}$ as shown in Fig. 7.

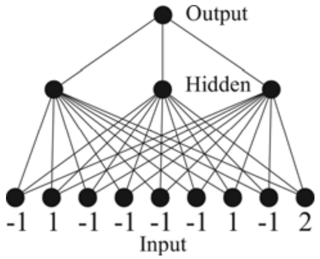


Fig. 7. ANN Structure

The training time of the ANN is 125.95 seconds. Fig. 8 shows the result by using GA and by using ANN. It shows a similar result on cannon distribution but gives a great improvement on run time as shown in Table 3.

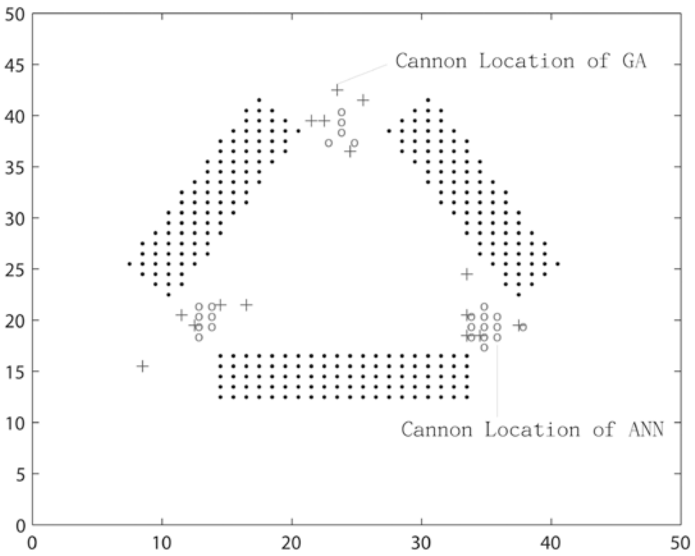


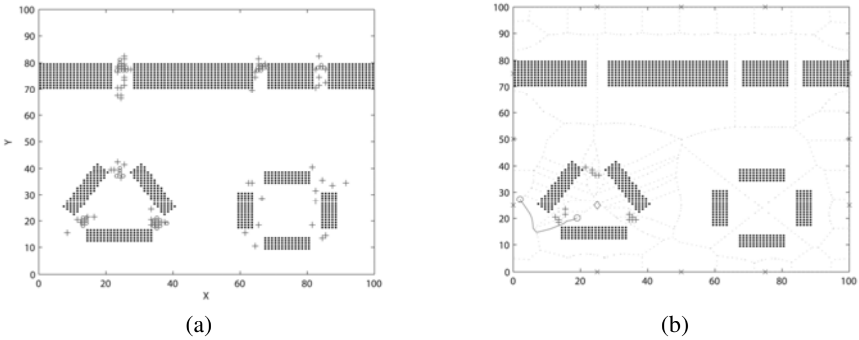
Fig. 8. Cannon distribution (Comparison between using GA and using ANN)

Table 3. Run time of ANN (Population size: 50, Generations: 150)

No. of Canyon	Run time of ANN (second)
1	2.81
2	2.89
3	3.18
4	3.43

4.3 Combine All Results into Single Battlefield

After ANN training, our machine learning component becomes very useful. In Fig. 9(a), it shows a battlefield that is commonly found in RTS games, (e.g., a number



(a)

(b)

Fig. 9. (a) Comparison between using GA and using ANN. (b) Result of using ANN.

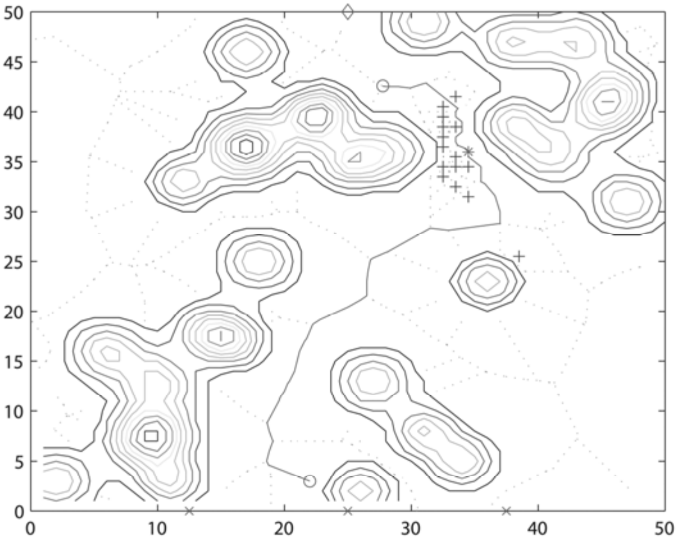


Fig. 10. Cannon distribution on random battlefield using ANN

of canyons in the landscape). The result of GA and ANN is similar but the run time of distributing cannons in this battlefield is only 21.266 seconds (in a Pentium IV environment). This component is also shown to be very useful in a more “real world like” battlefield as shown in Fig. 9(b) and Fig. 10.

5 Conclusion and Future Work

A neural-evolutionary model for case-based planning in real time strategy games is developed and shown in this paper. We believe that this research direction can provide an efficient, fair and natural AI development for RTS games. Base defense will only be part of our evaluations in our current and future works. We will extend the idea and combine with other key components in RTS games, such as resource management and battle strategies, in our future research.

Acknowledgement

This project is supported by the HK Polytechnic University Grant A-PA6N.

References

1. Lent, M.V., Laird, J.: Developing an Artificial Intelligence Engine. In: GDC Proceeding (1999)
2. Hsieh, J.L., Sun, C.T.: Building a player strategy model by analyzing replays of real-time strategy games. In: Neural Networks, IJCNN 2008 (2008)

3. Zanetti, S., Rhalibi, A.E.: Machine learning techniques for FPS in Q3. In: Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology, Singapore. ACM, New York (2004)
4. Cole, N., Louis, S.J., Miles, C.: Using a genetic algorithm to tune first-person shooter bots. In: Evolutionary Computation, Congress on Evolutionary Computation (2004)
5. Salge, C., Lipski, C., Mahlmann, T., Mathiak, B.: Using genetically optimized artificial intelligence to improve gameplaying fun for strategical games. In: Proceedings of the 2008 ACM SIGGRAPH symposium on Video games, Los Angeles, California. ACM, New York (2008)
6. Louis, S.J., Miles, C.: Playing to learn: case-injected genetic algorithms for learning to play computer games. IEEE Transactions on Evolutionary Computation 9(6), 669–681 (2005)
7. Holland, J.H.: Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT Press, Cambridge (1975)
8. Bourg, D.M., Seeman, G.: AI for Game Developers. O'Reilly Media, Inc., Sebastopol (2004)
9. Forbus, K.D., Mahoney, J.V., Dill, K.: How qualitative spatial reasoning can improve strategy game AIs. IEEE Intelligent Systems 17(4), 25–30 (2002)
10. Chuen-Tsai, S., Liao, Y.H., Lu, J.Y., Zheng, F.M.: Genetic algorithm learning in game playing with multiple coaches. In: Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence (1994)
11. Yi Jack, Y., Teo, J.: An Empirical Comparison of Non-adaptive, Adaptive and Self-Adaptive Co-evolution for Evolving Artificial Neural Network Game Players. In: 2006 IEEE Conference on Cybernetics and Intelligent Systems (2006)

Classification of Petroleum Well Drilling Operations with a Hybrid Particle Swarm/Ant Colony Algorithm

Adriane B.S. Serapião[1] and José Ricardo P. Mendes[2]

[1] UNESP/IGCE/DEMAC – C.P. 178 – CEP 13506-900 – Rio Claro (SP), Brazil
adriane@rc.unesp.br
[2] UNICAMP/FEM/DEP – C.P. 6122 – CEP 13081-970 – Campinas (SP), Brazil
jricardo@dep.fem.unicamp.br

Abstract. This paper describes an investigation of the hybrid PSO/ACO algorithm to classify automatically the well drilling operation stages. The method feasibility is demonstrated by its application to real mud-logging dataset. The results are compared with bio-inspired methods, and rule induction and decision tree algorithms for data mining.

1 Introduction

Motivated by rising drilling operation costs, the oil industry has shown a trend towards real-time measurements and control. In the last two decades, the technological advances in drilling techniques have notably contributed to the lowering of costs and to the expansion of exploration areas. On-line monitoring systems are as well gaining attention in drilling operations. Mud-logging is one of these systems, and it is used to measure and monitor mechanical and geological parameters.

The mud-logging system allows tracking the evolution of each measured parameter on-line. Nowadays, many parameters are continuously measured in the mud-logging units with a supervising purpose. However, in the most of cases, those data are stored without take advantage of all its potential. In other hand, to make use of the mud-logging data, it is required analyze and interpret it. That is not an easy task because of the large amount of information involved [1]. There is a lack of tools able to make an efficient use of all the data and information available.

This work presents the development of a system that intends to make better use of the information collected by mud-logging techniques during well drilling operations. The mud-logging techniques collected a great amount of data, and these data nowadays, are being used in a superficial way. The proposed system aims to take advantage of some of the information potential that is still not being used.

The proposed methodology is able to generate a precise report of the execution stages during an operation through the interpretation of mud-logging data. Potentially, the proposed methodology has two applications. The first is related to performance analysis and investigation of non-scheduled events. In this sense, the tool could carry out further analyses of the time spent on drilling each well in a field, and to determine how much time a given phase of drilling took as a fraction of total operational time. The second is related to the implementation of a computer system to produce an

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 301–310, 2009.

on-line report of the executed stages in the rig, and this report would present the precise time measurements as the mud-logging data.

Information concerning individual drilling performance can also be used to build benchmarking analysis. In this sense, a petrol company could use this information to compare the performance of different divisions. On a minor scale, the company could compare performance of rented rigs and identify weak points as part of ongoing improvement process. The results produced by an automatic classification system may help in the design of new wells. The information about the time spent to execute a determined stage could be used for planning new wells in the same region providing cost estimates.

Other hand, classification is one of the important data mining tasks. Data mining is a progress to extract implicit, nontrivial, previously unknown and potentially useful information (such as knowledge rules, constraints, regularities) from data in database. Data mining is an interdisciplinary field. The obtained results must conform to three main requisites: accuracy, comprehensibility and interest for the user [2].

Using Swarm Intelligence, in the context of data mining, for rule discovery is a quite new and challenging research area. In this work we investigate the use of the hybrid Particle Swarm Optimization/Ant Colony Optimization (PSO/ACO) algorithm to classify the well drilling operations using mud-logging data. PSO/ACO algorithm is a new tool for Data Mining, which discovers IF-THEN classification rules from data using continuous and nominal terms, without converting nominal values into numbers.

The hybrid PSO/ACO algorithm combines ideas from Ant Colony Optimization (Ant-Miner classification algorithm [2]) and Particle Swarm Optimization (Particle Swarm Data Mining Algorithms [3]) to create a classification meta-heuristic that supports both nominal and continuous attributes. The basic motivation for designing the hybrid algorithm is to make PSO more effective in coping with categorical attributes using the pheromone-based mechanism of ACO. The original algorithm proposed in [4] and [5], denoted PSO/ACO1, was designed to hierarchical classification. The new modified version of the PSO/ACO algorithm, hereafter denoted PSO/ACO2, uses a sequential covering approach to discover one-classification-rule-at-a-time [6].

The classification's results provided by PSO/ACO2 algorithm, corresponding to the stages of the drilling operation, were learnt based on a classification elaborated by a Petroleum Engineering expert [7] and then compared with others classification methods: bio-inspired algorithms (artificial immune systems (AIS): Parallel AIRS2 and CLONALG) [8], decision tree learning algorithms (J48, REPTree and LMT) and rule induction algorithm (JRIP) [9].

Many classification algorithms have been constructed and applied to discover knowledge from data in different applications, yet many suffer from poor performance in prediction accuracy in many practical domains. While it seems unlikely to have an algorithm to perform best in all the domains, it may well be possible to produce classifiers that perform better on a wide variety of real-world domains [2]. In this paper we have also two additional purposes. The first one is to study the contribution of recent natural computing algorithms (swarm and AIS algorithms) to solve real-world applications in the petroleum engineering area.

Rule and tree based methods are particularly recommended in applications where knowledge comprehensibility is very important, such as in engineering processes – where discovered knowledge should be carefully validated and interpreted by experts before they are actually used to diagnose a problem or suggest an intervention procedure, rather than blindly trust the result provided by an algorithm. The second purpose is to find the algorithm that leads to a best comprehensibility of the real problem, because it is very important topic whenever discovered knowledge will be used for supporting a decision made by a human user.

2 Mud-Logging System

During the petroleum well drilling operation many mechanical and hydraulic parameters are measured and monitored in order to perform the drilling in a safe and optimized manner. There are many systems that work together in a rig to accomplish this task. One of these systems is called a mud-logging system and it is responsible for measuring and monitoring a set of mechanical and geological parameters.

Mud-logging system techniques were introduced in Brazil in the 80's. At that time, only a reduced number of parameters were monitored. Since the 80s, with the developments in instrumentation techniques, the number of measured parameters has increased and the use of mud-logging systems became a common practice in the oil industry.

Deep and ultra-deep water drilling also contributed to the progress of mud-logging techniques in Brazil. Deep and ultra-deep water environments require very accurately controlled drilling operations. Any failure or negligence may cause human injury and economic losses. In order to have a more controlled process, the information supplier systems needed to be improved. In this context, the mud-logging systems were enhanced to become an important information supplier system.

Nowadays, mud-logging systems have two distinctive dimensions, the first one is responsible for collecting and analyzing formation samples (shale-shaker samples), and the second one is responsible for measuring and monitoring mechanical parameters related to the drilling operation. Considering only the second dimension, the mud-logging system could be characterized as a complete instrumentation system.

To accomplish its mission, the mud-logging systems rely in a wide range of sensors distributed in the rig operative systems. One important characteristic of this technique is that there is no sensor inside the well, and all measurements are taken on the rig. The data collect by the sensors are sent to a central computer system, where the data are processed and displayed in real time through screens installed in the mud-logging cabin and in the company-man office. The checking of the parameter evolution is carried out using the monitors. Throughout the whole drilling operation, there is a worker watching the parameters for any kind of abnormality.

The number of observed parameters may vary according to the particular characteristic of the drilling operation. The most common measured parameters are: Well Depth (Depth), True Vertical Depth (TVD), Bit Depth, Rate of Penetration (ROP), Hook Height, Weight on Hook (WOH), Weight on Bit (WOB), Vertical Rig Displacement (Heave), Torque, Drillstring Rotation per Minute (RPM), Mud Pit Volume, Pump Pressure, Choke Line Pressure, Pump Strokes per Minute (SPM), Stand Pipe Pressure (SPP), Mud Flow, Total Gas, Gas Concentration Distribution, H_2S Concentration, Mud Weight in/out, Drilling Fluid Resistivity, Drilling Fluid Temperature, Flow Line, LAG Time, and Stand Length.

3 Individual Stages Associated to the Drilling Operation for the Classification System

Well drilling is not a continuous process. On analysis it is noted that drilling is a sequence of discrete events. These events are referred to as ‘drilling operation stages’. Six basics stages associated to the well drilling operation were identified to build the proposed classification system:

- **Rotary Drilling:** The drilling itself occurs; the bit really advances increasing well depth. The drill string is rotating and there is mud circulation. The drill string is not anchored to the rotary table causing a high hook weight level.
- **Rotary Reaming:** Despite the high hook weight level, mud circulation and drillstring rotation, the bit does not advance increasing the final well depth. In this situation, there is a back-reaming of an already drilled well section.
- **Oriented Drilling (“Sliding Drilling”):** The bit really advances increasing the well depth. The difference here is that the drillstring is not rotating and the drilling occurs due to the action of the downhole motor. There is mud circulation and a high hook weight level.
- **Back-reaming or Tool adjusting:** The bit does not advance increasing the final well depth. There is circulation and a high hook weight level. This condition indicates that back-reaming is being carried out or that the tool-face of the downhole tool is being adjusted.
- **Tripping:** This stage corresponds to the addiction of a new section to the drillstring. It is characterized when there is no well depth variation, no drillstring rotation, and there is low hook weight and low pressure levels in the stand pipe.
- **Circulating:** There is no gain in the well depth. It is characterized by fluid circulation, a high hook weight level and a moderated rotation of the drillstring.

These six stages represent a first effort to individualize the basic components of a drilling operation. The stages were detailed considering the drilling phases with mud return to the surface. The drilling technology considered was the drilling using mud motor and bent housing. This classification may not be satisfactory for the initial drilling phases and for special operations, such as fishing, in the well. In the same way, if other drilling technologies are considered, like the rotary steerable systems, small adjustments in the definition of the stages will be required. For instance, when using rotary steerable systems, it makes no sense to make a distinction between rotary drilling and oriented drilling stages as they were defined in this work, because these systems are supposed to drill all the time using drillstring rotation.

In order to identify the stage that in being executed, the system need some of the information monitored by the mud-logging systems (classifier’s inputs). This work suggests the use of five parameters:

- **Bit Depth:** This parameter measures the current bit depth.
- **Weight on Hook (WOH):** It measures the current load level supported by the hook in a time unit.
- **Drillstring Rotation (RPM):** This parameter measures the revolutions carried out by the drillstring.

- **Stand Pipe Pressure (SPP):** This parameter measures the mud flow pressure level in the stand pipe to the well.
- **Weight on Bit (WOB):** This parameter represents the load that the bit applies to the rock formation during the drilling operation.

4 The Hybrid PSO/ACO Algorithm

PSO/ACO2 is an algorithm proposed for the discovery of classification rules in data mining problems. The algorithm works with continuous and nominal attributes. A rule consists of an antecedent (a set of attribute-values) and a consequent (class). The consequent of the rule is the class that is predicted by that rule. The antecedent consists of a set of terms. A term is defined by a triple *<attribute, operator, value>*, where value is a value belonging to the domain of attribute. The operator used is "=" in the case of categorical/nominal attributes, or "≤" and ">" in the case of continuous attributes. In this context, each case is assigned to one predefined class according to the values of its attributes for the case. The discovered knowledge is expressed in the form of IF-THEN rules (Eq.1), as follows:

$$\text{IF } \langle \text{attrib} = \text{value} \rangle \text{ AND } \ldots \text{ AND } \langle \text{attrib} \geq \text{value} \rangle \text{ THEN } \langle \text{class} \rangle \tag{1}$$

Here we provide just an overview of the Particle Swarm Optimization / Ant Colony Optimization (PSO/ACO2) algorithm used in our experiments. The algorithm is proposed and described by Holden and Freitas in [6]. For details, consult the references [3, 4, 5, 6]. PSO/ACO2 algorithm uses a sequential covering approach to discover one-classification-rule-at-a-time, as shown in Pseudocode 1.

Pseudocode 1. Sequential Covering Approach used by PSO/ACO2 Algorithm

```
$RS = \emptyset$ /* Discovered Rule Set. Initially, RS is empty */
FOR EACH class C
  TS = {All training examples}
  WHILE (Number of uncovered training examples of the class C > MaxUcovExampPerClass)
    Run the PSO/ACO algorithm to discover the best nominal rule predicting class C, called Rule
    Run the standard PSO algorithm to add continuous terms to Rule, and return the best discovered rule BestRule
    Prune BestRule
    $RS = RS \cup BestRule$
    $TS = TS -$ {training examples correctly covered by discovered rule}
  END WHILE
END FOR
Order rules in RS by descending Quality  /* higher quality measures */
```

The rule-quality is used to evaluate the predictive accuracy of a candidate rule with the *Precision* formula [6], based on its instances classified as true and false positives.

5 Results and Discussion

The classification problem consists in identifying the drilling operations described above as *Rotary Drilling* (RD), *Rotary Reaming* (RR), *Oriented Drilling ("Sliding Drilling")* (SD), *Back-reaming or Tool adjusting* (TA), *Tripping* (TR) and *Circulat-*

ing (CI). The mud-logging parameters selected to represent each class (attributes) in the classification task of a given drilling stage in execution are: Bit Depth, Weight on Hook (WOH), Stand Pipe Pressure (SPP), Drillstring Rotation (RPM) and Weight on Bit (WOB), as in our previous work [7], as described in the Section 3. Others parameters were not used because they are not significant for the drilling classification.

Table 1. Distribution of data per class in the training and test sets

Number of samples	**Drilling operations**						
	CI	**TR**	**TA**	**SD**	**RR**	**RD**	**Total**
Training Set	14	75	795	753	343	858	*2838*
Test Set	2	22	266	253	114	289	*946*

Real records of mud-logging data consisting of 3784 samples of three days well drilling were used for the training and evaluation of PSO/ACO2 algorithm. A Petroleum Engineering expert classified previously these data [7]. We have splitted arbitrarily the data set into a learning set (2838 samples – 75% of the original set), which is used for the computation of rules in the PSO/ACO2 algorithm, and a test set (946 samples – 25% of the original set), which is used for the model performance evaluation. These sets were used previously in the others related classification methods for this problem [7]. Table 1 shows the data distribution according to predefined classes for the training and test sets. The table clearly indicates the data imbalance issue among the classes, mainly for the Circulating (CI) and Tripping (TR) stages, which are the less usual operation in the drilling activity.

Experiments for classification with swarm intelligence techniques (hybrid PSO and ACO) were conducted using PSO/ACO2 software developed by Nicholas Holden and provided in the SourceForge[1] project site. Upon experimentation, the suitable numbers of particles and iterations were found to be both 100 and the number of uncovered examples per class (*MaxUncovExampPerClass*) to 2. For PSO algorithm the constriction factor is $\chi = 0.72984$, and cognitive and social learning coefficients are $c_1 = c_2 = 2.05$. Also, the rule-quality measure (fitness function) was precision (Eq.2). The results of this swarm algorithm were compared with others nature-based and classification rule mining approaches described as follows.

The algorithms' implementations of artificial immune systems (CLONALG and Parallel AIRS2) were developed in Java language. For the proposed task CLONALG was trained with 20 generations. Parallel AIRS2's experiments are undertaken with the *k*-value for the *k* nearest neighbor approach is set to 7. The value for number of threads is 5. Details about these algorithms and about the results of their application in drilling operation classification can be found in our previous work [8].

For the Machine Learning experiments, we used well-known tree and rule based classification algorithms available in the Waikato Environment for Knowledge Analysis (WEKA) software package [9]. The algorithms are as follows. J48 is an implementation of the well-known Quinlan algorithm (C4.5). This classifier builds a decision tree whose nodes represent discrimination rules acting on selective features. REPTree (Reduced Error Pruning Tree) can build a decision or regression tree using information gain as the

[1] Available at http://sourceforge.net/projects/psoaco2/

splitting criterion and prunes trees using reduced-error pruning. LMT (Logistic Model Tree) is a data mining algorithm for building logistic model trees, which are classification trees with logistic regression functions at the leaves. JRIP implements Repeated Incremental Pruning to Produce Error Reduction (RIPPER), a prepositional rule learner. Rules are created for every class in the training set and are then pruned.

In these algorithms, the discovered knowledge is represented in the form of IF-THEN prediction rules which have the advantage of being a high-level and symbolic knowledge representation contributing towards the comprehensibility of the discovered knowledge.

In all the experiments reported, we have used a 10-fold cross-validation procedure.

The hybrid particle swarm optimization/ant colony optimization algorithm produced in the training phase around 29 classification rules used to identify the drilling stages. Table 2 shows the classification accuracy (contingency table) of PSO/ACO2 algorithm for training and test sets, in the classification task of well drilling operations, using the settings parameters mentioned earlier and the discovered rules. The PSO/ACO2 algorithm was able to achieve a success rate of 92.53% in the training set and 94.08% in the test set.

Table 2. Contingency tables for the training and test sets

a) Training set

		Estimated classification					
		CI	**TR**	**TA**	**SD**	**RR**	**RD**
Real classification	**CI**	13	0	1	0	0	0
	TR	0	75	0	0	0	0
	TA	0	0	710	85	0	0
	SD	0	0	6	747	0	0
	RR	1	0	0	0	251	91
	RD	1	0	0	0	27	830

b) Test set

		Estimated classification					
		CI	**TR**	**TA**	**SD**	**RR**	**RD**
Real classification	**CI**	2	0	0	0	0	0
	TR	0	22	0	0	0	0
	TA	0	0	240	26	0	0
	SD	0	0	0	253	0	0
	RR	0	0	0	0	89	25
	RD	0	0	0	0	5	284

We have also investigated PSO/ACO2's robustness to the parameter settings: number of iterations, number of particles and number of folds, when a parameter changed and the others were fixed. In this application, the performances for different iterations (50, 100, 200 and 300) were very small (around 1% between the worst and the best result) after extensive simulations. Varying the number of particles among 25 and 1024, the difference between the lowest and the highest success rate of all experiments was less than 0.8%. Changing the number of folds (from 2 to 20) caused the greatest impact in the correctness classification (maximum of 1.3% between the worst and the best performance). The algorithm seems to be quite robust to many combinations of parameter settings, according to a large number of performed simulations.

The performance among all algorithms was compared using four criteria: (1) percentage of success in the classification task; (2) solution quality, the learning of discrimination of classes; (3) number of partitions (rules/leaves/antibodies) generated for the classifier; (4) the processing time for training. Number of partitions is the way where we compared the simplicity of the discovered knowledge by the number of discovered rules. The average number of generated rules per fold during the training phase was considered. The processing time, and not the number of generation cycles (iterations), was used to measure the speed of each method, because the number of iterations is different from one algorithm to another.

To make the comparison as fair as possible, all algorithms used exactly the same training and test set as discussed earlier. Tables 3 and 4 present a comparative framework of the classification accuracy for each class related to each learning technique for both training and test data, respectively. Table 5 summarizes the performance of each algorithm in terms of number of partitions and processing time. All experiments took place on a 3.0 GHz Pentium IV machine.

Table 3. Classification accuracy for each class in the training set

Method	**Correct classification of the drilling operations**						**Overall accuracy**
	CI	**TR**	**TA**	**SD**	**RR**	**RD**	
PSO/ACO2	92.9%	100%	89.3%	99.2%	73.2%	96.7%	92.53%
J48	100%	98.7%	89.2%	99.7%	74.3%	98.4%	93.27%
REPTree	92.9%	100%	89.3%	99.3%	73.8%	98.0%	93.02%
LMT	92.9%	100%	89.1%	99.7%	74.6%	98.0%	93.16%
JRIP	64.3%	98.7%	89.3%	99.5%	74.3%	97.9%	92.92%
CLONALG	0%	98.7%	85.0%	93.5%	45.5%	96.5%	85.91%
Parallel AIRS2	57.1%	100%	90.1%	95.6%	69.7%	96.6%	91.16%

Table 4. Classification accuracy for each class in the test set

Method	**Correct classification of the drilling operations**						**Overall accuracy**
	CI	**TR**	**TA**	**SD**	**RR**	**RD**	
PSO/ACO2	100%	100%	90.2%	100%	78.1%	98.3%	94.08%
J48	100%	100%	90.2%	100%	86.0%	97.6%	94.82%
REPTree	100%	100%	90.2%	100%	78.1%	98.3%	94.08%
LMT	100%	100%	90.2%	100%	83.3%	97.2%	94.40%
JRIP	100%	100%	90.2%	100%	82.5%	96.5%	94.08%
CLONALG	0%	100%	89.5%	96.0%	48.2%	98.3%	89.01%
Parallel AIRS2	100%	100%	91.7%	97.6%	75.4%	98.3%	92.92%

According to Tables 3 and 4, PSO/ACO2 algorithm outperformed CLONALG and Paralell AIRS2 in all criteria, except the time processing for the Parallel AIRS2, whereas it was slightly less successful than all rule- and tree-based algorithms for success rate, solution quality and processing time. PSO/ACO2 is better than the AIS algorithms because of its ability to discriminate the minority classes appropriately, in contrast to the weakness of immune algorithms for this case. In addition, the number of partitions generated by AIS algorithms (upper 600 antibodies) is very superior to the 29 rules generated by PSO/ACO2 algorithm, as indicated in Table 5.

While PSO/ACO2 algorithm was able to achieve a similar performance for training data, the same success rate (94.08%) and the same solution quality for unseen data (test set) as REPTree, the number of rules is greater and the processing time to build the classifier had a longer duration than REPTree. It is almost the same case for comparing PSO/ACO2 with JRIP. Other hand, PSO/ACO2 and LMT have a similar time processing, although the performance of PSO/ACO2 is slightly lower.

The J48 method was found to perform better than other algorithms in terms of success rate, solution quality and processing time, whilst being fifth best in terms of number of partitions. J48 led to the discovery of rules considerably larger (and so more difficultly interpretable by users) than the rules discovered with the others classification rules mining methods, including the PSO/ACO2 meta-heuristic, as indicated in Table 5. That can be considered a potential disadvantage of this method.

Still with respect to the comprehensibility of the discovered rules, JRIP presented the smaller number of partitions. A simpler rule set facilitates the interpretation of the discovered rules, and therefore is a desirable result.

Regarding the processing time, CLONALG processes the data in a computationally expensive way. We can see that J48 and REPTree's running time is fewer than all other methods.

Table 5. Comparing number of rules and processing time

Method	Number of partitions (rules-leaves-antibodies)	Processing time
PSO/ACO2	29 rules	7'46"
J48	24 leaves (= 47 rules)	0'02"
REPTree	5 leaves (= 11 rules)	0'02"
LMT	10 leaves (=19 rules)	7'45"
JRIP	10 rules	0'17"
CLONALG	706 antibodies	40'09"
Parallel AIRS2	610 antibodies	0'25"

6 Conclusion

This work shows the practical suitability of PSO/ACO2 for application to classification task. The paper shows that classification using PSO/ACO2 algorithm is better than classifiers based on artificial immune systems (Parallel AIRS2 and CLONALG). The imbalanced real mud-logging data has a large impact on the classification performance of the AIS classifiers, since they can achieve high precision on predominant classes but very low correct classification on classes with a few samples. PSO/ACO2 algorithm seems to contour this effect.

The experiments show that, in general, the PSO/ACO2 algorithm is at least comparable in terms of accuracy with the rule and tree learners algorithms, J48, REPTree, LMT and JRIP. Quantitative comparison of PSO/ACO2 algorithm versus these methods indicates that the PSO/ACO2 was especially less capable of discriminating rotating mode classes (RR and RD) than these others approaches, producing so a lower correct classification rate.

PSO/ACO2 algorithm is originally proposed to deal with nominal attributes without converting its values to a binary representation and later to combine with continuous attributes. In this paper we are considering only continuous values. We believe this algorithm can be improved with some adjustments to increase its computational effort for applications like the presented here, so as to converge faster. The predictive accuracy of PSO/ACO2 algorithm is not so sensitive to values of the parameters settings. PSO/ACO2 proved to be a reliable technique for real data multi-class classification.

Because the application of swarm meta-heuristics in data mining, especially in classification rule mining, is still in infant periods, in future works, our further research will involve experiments with other kinds of petroleum engineering data and with mixed nominal/continuous attributes. Second, we will further be interesting to investigate the performance of other kinds of heuristic function to solve real world problems.

Acknowledgements

The authors would like to acknowledge Nicholas Holden of University of Kent at Canterbury for his technical support, improvements and adjustments in the software for this work.

References

1. Kyllingstad, A., Horpestad, J.L., Klakegg, S., Kristiansen, A., Aadnoy, B.S.: Factors Limiting the Quantitative Use of Mud-Logging Data. In: Proc. of the SPE Asia Pacific Oil and Gas Conference, Singapore (1993)
2. Parpinelli, R.S., Lopes, H.S., Freitas, A.A.: Data Mining with an Ant Colony Optimization Algorithm. IEEE Trans. on Evolutionary Computation, special issue on Ant Colony algorithms 6(4), 321–332 (2002)
3. Sousa, T., Silva, A., Neves, A.: Particle Swarm based Data Mining Algorithms for classification tasks. Parallel Computing 30, 767–783 (2004)
4. Holden, N., Freitas, A.A.: A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data. In: Proc. of 2005 IEEE Swarm Intelligence Symposium, Pasadena, California, pp. 100–107 (2005)
5. Holden, N., Freitas, A.A.: Hierarchical classification of G-protein-coupled receptors with a PSO/ACO algorithm. In: Proc. of 2006 IEEE Swarm Intelligence Symposium, Indianapolis, pp. 77–84 (2006)
6. Holden, N., Freitas, A.A.: A Hybrid PSO/ACO Algorithm for Classification. In: Proc. of Genetic and Evolutionary Computation Conference (GECCO 2007), London (2007)
7. Tavares, R.M., Mendes, J.R.P., Morooka, C.K., Plácido, J.C.R.: Automated Classification System for Petroleum Well Drilling using Mud-Logging Data. In: Proc. of 18th International Congress of Mechanical Engineer, Offshore & Petroleum and Engineering, Ouro Preto, Brazil (2005)
8. Serapião, A.B.S., Mendes, J.R.P., Miura, K.: Artificial Immune Systems for Classification of Petroleum Well Drilling Operations. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 395–406. Springer, Heidelberg (2007)
9. Witten, I., Frank, M.: Data Mining: Practical Machine Learning Tool and Technique with Java implementation. Morgan Kaufmann, San Francisco (2000)

A Robust Technique for Background Subtraction and Shadow Elimination in Traffic Video Sequence

Tao Gao[1], Zheng-guang Liu[1], Wen-chun Gao[2], and Jun Zhang[1]

[1] School of Electrical Engineering and Automation, Tianjin University,
Tianjin, 300072, China
[2] Honeywell (China) Limited, Tianjin, 300042, China
gaotao09@yahoo.cn

Abstract. A novel background model based on Marr wavelet kernel and a background subtraction technique based on binary discrete wavelet transforms are introduced. The background model keeps a sample of intensity values for each pixel in the image and uses this sample to estimate the probability density function of the pixel intensity. The density function is estimated using a new Marr wavelet kernel density estimation technique. Since this approach is quite general, the model can approximate any distribution for the pixel intensity without any assumptions about the underlying distribution shape. The background and current frame are transformed in the binary discrete wavelet domain, and background subtraction is performed in each sub-band. After obtaining the foreground, shadow is eliminated by an edge detection method. Experiments show that the simple method produces good results with much lower computational complexity and can effectively extract the moving objects, even though the objects are similar to the background, and shadows can be successfully eliminated, thus good moving objects segmentation can be obtained.

Keywords: Background modeling, background subtraction, Marr wavelet, BDWT, shadow elimination.

1 Introduction

Identifying moving objects from a video sequence is a fundamental and critical task in many computer-vision applications. Background subtraction [1, 2] is a method typically used to detect unusual motion in the scene by comparing each new frame to a model of the scene background. In traffic video surveillance systems, stationary cameras are typically used to monitor activities on the road. Since the cameras are stationary, the detection of moving objects can be achieved by comparing each new frame with a representation of the scene background. This process is called background subtraction and the scene representation is called the background model. Typically, background subtraction forms the first stage in automated visual surveillance systems. Results from background subtraction are used for further processing, such as tracking targets and understanding events. A mixture of Gaussians is used in [3] to model each pixel's intensity. The models are learned and updated for

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 311–322, 2009.

each pixel separately. A mixture of three normal distributions is used in [4] to model the pixel value for traffic surveillance applications. The pixel intensity is modeled as a weighted mixture of three normal distributions: road, shadow and vehicle distribution. An incremental EM algorithm is used to learn and update the parameters of the model. In [5, 6] kalman and wiener filtering are performed at every pixel. Foreground is detected when the observed intensity is different than the predicted intensity. Ref. [7] modifies the background image that is subtracted from the current image so that it looks similar to the background in the current video frame. The background is updated by taking a weighted average of the current background and the current frame of the video sequence. In this paper, we propose a new model for background maintenance and subtraction. A sample of intensity values for each pixel is kept and used to estimate the Marr wavelet probability density function of the pixel intensity. After background modeling, binary discrete wavelet transforms is used for background subtraction. In our experiments about the video surveillance on urban road, the model can solve the problem of gradual change of illumination and it can detect both moving vehicles and foot passengers.

2 Previous Background Modeling Methods

Background modeling is at the heart of any background subtraction algorithm. Several models have been put forward for background maintenance and subtraction described in introduction. In this paper, we focus only on the two most commonly used techniques, and exclude those which require significant resource for initialization or are too complex.

2.1 Frame-Difference Background Modeling

Frame-difference method [7, 8, 9] obtains the background image as follows:

$$BW_i = \begin{cases} 1 & if \quad abs(I_i - I_{i-1}) \geq A \\ 0 & if \quad abs(I_i - I_{i-1}) < A \end{cases} \tag{1}$$

The histogram of the difference image will have high values for low pixel intensities and low values for the higher pixel intensities. To set the threshold A, a dip is looked for in the histogram that occurs to the right of the peak. Starting from the pixel value corresponding to the peak of the histogram, we search toward increasing pixel intensities for a location on the histogram that has a value significantly lower than the peak value (using 10% of the peak value). The corresponding pixel value is used as the new threshold A. The background B_i:

$$B_i = \begin{cases} B_{i-1}(x, y) & BW_i(x, y) = 1 \\ aI_i + (1-a)B_{i-1}(x, y) & BW_i(x, y) = 0 \end{cases} \tag{2}$$

Then B_i is the background image. The weight a assigned to the current and instantaneous background affect the update speed, empirically determined to be 0.1.

2.2 Mixture of Gaussians Background Modeling

In Mixture of Gaussians (MoG) [10], each pixel location is represented by a number (or mixture) of Gaussians functions that sum together to form a probability distribution function F :

$$F(i_t = \mu) = \sum_{i=1}^{k} \omega_{i,t} \cdot \eta(\mu, \sigma) \tag{3}$$

To determine if a pixel is part of the background, we compare the input pixels to the means μ_i of their associated components. If a pixel value is close enough to a given component's mean, that component is considered a matched component. Specifically, to be a matched component, the absolute difference between the pixel and mean must be less than the component's standard deviation scaled by a factor $D : \left| i_i - \mu_{i,t-1} \right| \leq D \cdot \sigma$. Then we update the component variables (ω, μ , and σ) to reflect the new pixel value. For matched components, a set of equations increase our confidence in the component (ω increases, σ decreases, and μ is nudged towards the pixel value). For non-matched components, the weights decrease exponentially (μ and σ stay the same). How fast these variables change is dependent on a learning factor p present in all the equations. Then we determine which components are parts of the background model. First, we order the components according to a confidence metric ω / σ, which rewards high ω and low σ . We do this because we want to keep only the M most confident guesses. Second, we apply a threshold to the component weights ω. The background model is then the first M components (in order of highest to lowest ω / σ), whose weight ω is above the threshold. M is the maximum number of components in the background model, and reflects the number of modes we expect in the background probability distribution function F . Last, we determine foreground pixels. Foreground pixels are those that don't match any components determined to be in the background model.

3 A Robust Background Modeling Method

3.1 Marr Wavelet

Marr wavelet [11] is the second derivative of Gaussian smooth function $\psi_m = C_m (-1)^m \frac{\mathrm{d}^2}{\mathrm{d}t^m} (\mathrm{e}^{-\frac{t^2}{2}}) |_{m=2}$, that is:

$$\psi(t) = \frac{2}{\sqrt{3}} \pi^{-1/4} (1 - t^2) \mathrm{e}^{-t^2/2} \tag{4}$$

The coefficient $\frac{2}{\sqrt{3}} \pi^{-1/4}$ is a guarantee of normalization of $\psi(t)$: $\|\psi\|^2 = 1$. Marr wavelet is widely used in visual information processing, edge detection and other fields. Due to the obvious advantages of extracting characters of the background (especially the edge), this paper uses Marr wavelet function instead of traditional Gaussian function to model the background. The difference of Gaussians (DOG) is a good approximation to Marr wavelet, so we use DOG to approximate Marr wavelet:

$$\psi(t) = \mathrm{e}^{-\frac{t^2}{2}} - \frac{1}{2}\mathrm{e}^{-t^2/8} \tag{5}$$

3.2 Background Modeling

The first frame can be set as the initial background. Considering the dithering of camera, the bias matrix should be obtained before background modeling. We mainly concern the up-down, left-right plane dithering. If the background is B and current frame is f , they are firstly processed with two or three level Gaussian pyramid decomposition:

$$f_{n-1} = (f_n \otimes \frac{e^{-(n_1^2+n_2^2)/(2\sigma^2)}}{\sum_{n_1}\sum_{n_2} e^{-(n_1^2+n_2^2)/(2\sigma^2)}}) \downarrow_2 \tag{6}$$

Where $n_1 = n_2$ =3, and σ is 0.5. Supposing the bias matrix is $[v, h]$, v and h representative the vertical and level dithering parameters. If the size of f_{n-1} is $M \times N$, the n-1 level initial bias matrix is $[v_{n-1}, h_{n-1}]$, and the maximum pixel value is $f_{\max}$, minimum is $f_{\min}$, so the bias function is as follows:

$$\begin{aligned} Mo = \frac{\alpha}{\sqrt{2\pi\sigma}} | e^{-(f_{n-1}-B_{n-1,v_{n-1},h_{n-1}}(\Delta x,\Delta y))^2/2\sigma} \\ -\frac{1}{2}e^{-(f_{n-1}-B_{n-1,v_{n-1},h_{n-1}}(\Delta x,\Delta y))^2/8\sigma} | + (1-\alpha)\frac{1}{f_{\max}-f_{\min}} \end{aligned} \tag{7}$$

Where α is 0.3, and σ is $\beta \cdot (f_{\max} - f_{\min})^2$, β =0.01, $(\Delta x, \Delta y)$ is the offset distance. The $(\Delta x, \Delta y)$ is changed to obtain the maximum value Ds according to $Ds = \frac{\log(Mo)}{\sum_M \sum_N f_{n-1}}$, and thus to get the best offset distance $(\Delta x_{best}, \Delta y_{best})$. The $[v_{n-1} + \Delta x_{best}, h_{n-1} + \Delta y_{best}]$ is used as the initial bias matrix for next level f_n . By the iterative process, the dithering distance can be finally determined. If the original background after offset correction is $B(i, j)$, and current frame is $f(i, j)$, we define the probability distribution of deviation between background and current frame is:

$$PS(i, j) = \frac{\alpha}{\sqrt{2\pi\sigma}} \left| e^{-\frac{(B(i,j)-f(i,j))^2}{2\sigma_0}} - \frac{1}{2}e^{-\frac{(B(i,j)-f(i,j))^2}{8\sigma_0}} \right| \tag{8}$$

The updating weight for background pixel is as follows:

$$\Delta S(i,j)=\frac{PS(i,j)}{(PS(i,j)+(1-\alpha)\frac{1}{f_{\max}-f_{\min}})} \tag{9}$$

The iterative process for updating background is:

$$\sigma_i=\frac{\sum_{n=1}^{N}\Delta S_{n,i-1}\cdot(B_{i-1}-f_n)^2}{\sum_{n=1}^{N}\Delta S_{n,i-1}} \quad B_i=\frac{\sum_{n=1}^{N}\Delta S_{n,i}\cdot f_n}{\sum_{n=1}^{N}\Delta S_{n,i}} \tag{10}$$

Where i is the iteration number, and N is the frame number. Then B_i is the final background.

4 BDWT Based Background Subtraction

For one dimension signal $f(t)$, binary discrete wavelet [12] transforms are as follows:

$$S_{2^j}f(t)=f(t)*\phi_{2^j}(t) \tag{11}$$

$$W_{2^j}f(t)=f(t)*\varphi_{2^j}(t) \tag{12}$$

$S_{2^j}f(t)$ is the projection of $f(t)$ in the V_j space, and $W_{2^j}f(t)$ is the projection of $f(t)$ in the W_j space. Equation (11) (12) can be rewritten as:

$$S_{2^j}f(t)=\sum_{l\in Z}S_{2^{j-1}}f(t-l)h_{j-1}(l) \tag{13}$$

$$W_{2^j}f(t)=\sum_{l\in Z}S_{2^{j-1}}f(t-l)g_{j-1}(l) \tag{14}$$

For a digital signal $d(n)=S_1f(t)|_{t=nT}=S_1f(n)$,

$$S_{2^j}f(n)=\sum_{l\in Z}S_{2^{j-1}}f(n-l)h_{j-1}(l)\quad j=1,2,\cdots \tag{15}$$

$$W_{2^j}f(n)=\sum_{l\in Z}S_{2^{j-1}}f(n-l)g_{j-1}(l)\quad j=1,2,\cdots \tag{16}$$

Where $H(z)=\frac{1}{4}\left(-\frac{1}{2}z^{-2}+z^{-1}+3+z-\frac{1}{2}z^{2}\right)$ and $G(z)=\frac{1}{4}\left(z^{-1}-2+z\right)$ ·
Because the coefficients of the sub-bands of the BDWT are highly correlated, and the direction and size are the same as the image, also, there is no translation in the sub-bands; we define the difference between digital signal $d_1(n)$ and $d_2(n)$ as:

$$DE=\sum_{j=J_0}^{J_1}\{\left|S_{2^j}f_1(n)-S_{2^j}f_2(n)\right|+\left|W_{2^j}f_1(n)-W_{2^j}f_2(n)\right|\} \tag{17}$$

The J_0 and J_1 are the starting and ending scales. For a two dimensions digital image, we perform the BDWT to the rows and then the columns. After acquiring the *DE* between two frames, the motion area is obtained by setting a threshold which can be obtained automatically by otsu [13] method. The noise can be removed by mathematical morphology method. In application, one of the two frames is a background image; the other is the current frame. Figure.1 shows the comparison of segmentation results between our method and frame difference.

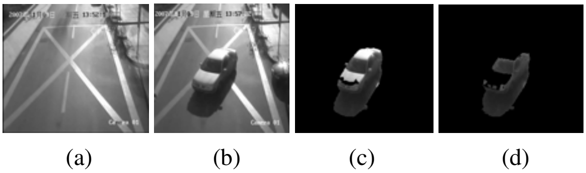


Fig. 1. The comparison of motion areas (a) Background (b) Original image (c) Motion area obtained by BDWT (d) Motion area obtained by frame-difference

5 Shadow Removing

In actual test, shadows extracted along with the objects can result in large errors in object localization and recognition. We propose a method of moving shadow detection based on edge information, which can effectively detect the cast shadow of a moving vehicle in a traffic scene. In practice, edge detection typically involves the estimation of first and perhaps second derivatives of the luminance function, followed by selection of zero crossings or extrema. For natural images, most shadow scenes are relatively smooth in areas between edges, while large discontinuities occur at edge locations of the moving object. So shadow can be seen as the sensor noise.

In this paper, edge detection depends upon the reliable detection of a non-zero gradient in the luminance function, and the reliable inference of the sign of the second derivative of the luminance function in the gradient direction [14]. The gradient estimates are computed using the two basis functions:

$$g_1^x(x,y,\sigma_1)=\frac{-x}{2\pi\sigma_1^4}e^{-(x^2+y^2)/2\sigma_1^2} \tag{18}$$

$$g_1^y(x,y,\sigma_1)=\frac{-y}{2\pi\sigma_1^4}e^{-(x^2+y^2)/2\sigma_1^2} \tag{19}$$

The second derivative estimates are computed using the three basis functions:

$$g_2^x(x,y,\sigma_2)=\frac{1}{2\pi\sigma_2^4}((x/\sigma_2)^2-1)e^{-(x^2+y^2)/2\sigma_2^2} \tag{20}$$

$$g_2^y(x,y,\sigma_2)=\frac{1}{2\pi\sigma_2^4}((y/\sigma_2)^2-1)e^{-(x^2+y^2)/2\sigma_2^2} \tag{21}$$

$$g_2^{xy}(x,y,\sigma_2)=\frac{-y}{2\pi\sigma_2^6}e^{-(x^2+y^2)/2\sigma_2^2} \tag{22}$$

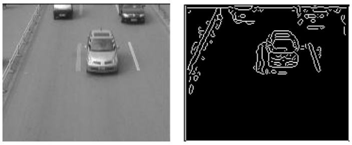
where σ_1 denotes the scale of the first derivative smoothing kernel, and σ_2 denotes the scale of the second derivative smoothing kernel. We use a multi-scale approach where different scales of sigma are employed: $\sigma_1 \in [2,4,8]$ and $\sigma_2 \in [1,2,4]$ pixels, so sharp images with little noise require a small scale narrow filter while blurry noise images require a large scale filter. If the current frame is im, the gradient angle α is:

$$\alpha = angle(g_1^x \otimes im + i \cdot g_1^y \otimes im) \tag{23}$$

The threshold $T\alpha = ni \cdot \frac{1}{2\sqrt{2\pi}\sigma_1^2} \cdot \sqrt{(-2)\cdot \log \beta}$, where $\beta = 2 \cdot e^{-6}$, if $|\alpha| < T\alpha$, then $\alpha = 0$. The lap of im at the angle α is:

$$lap = \cos^2\alpha \cdot g_2^x \otimes im + \sin^2\alpha \cdot g_2^y \otimes im - 2\cdot\cos\alpha\cdot\sin\alpha\cdot g_2^{xy} \otimes im \tag{24}$$

Threshold $Tl = ni \cdot \cfrac{\frac{\sqrt{2}}{(1-\beta)\sqrt{(1-\beta)}}}{4\sqrt{\pi/3}\cdot\sigma_2^3}$, if $|lap| < Tl$, then $lap = 0$, the sensor noise $ni = 10$. For the lap image, edges can be obtained by ‘Canny’ operator. Figure.2 shows the edge detection after removing shadow edge. We use eight directions assimilation method [15, 16] to fill the edges, combined with the foreground information.



current frame edge detection

Fig. 2. Edge detection without shadow edge

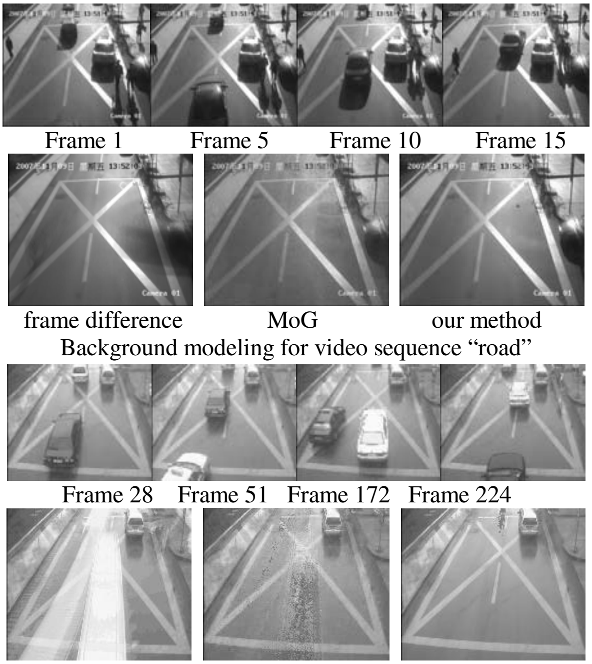
6 Experimental Results

In this section, we compare the performance of some background modeling techniques: frame-difference, MoG, and our method with uses BDWT based motion segmentation to perform background subtraction. The video sequences used for testing were taken from urban traffic monitoring cameras. The video was sampled at a resolution of 768x576 and a rate of 15 frames per second. Once the frames were loaded into memory, our algorithm averaged 18 frames per second on a 1400 MHz Celeron CPU. Although we used grayscale video as input, and output a grayscale background model, it is straightforward to extend the algorithm to use color images.

We use the error measurement quantify how well each algorithm matches the ground-truth. It is defined in our context as follows:

$$error = \frac{\left|\text{Number of foreground pixels identied by the algorithm - Number of foreground pixels in ground-truth}\right|}{\text{Number of foreground pixels in ground-truth}}$$

Figure.3 shows four sample frames of each test video, and the background modeling results for every video sequence. The frame number of video "road" and "car" is respectively 160 and 224. Figure.4 shows the background subtraction by our method. Figure.5 shows background subtraction by frame-difference, and Figure.6 shows background sub-traction by MoG. Figure.7 and Figure.8 shows the error measurement of each method for video sequence "road" and "car". From the experiments we can see that the frame-difference method has two drawbacks: first, as the foreground objects may have a similar color as the background, these objects can not be detected by threshold. Second, the method is only slowly adapting to slightly changing environmental conditions. Thus, faster changes as a flashlight signal or fluttering leaves in the wind can not be modeled. Also, in practice, modeling the background variations with a small number of Gaussian distribution will not be accurate. Furthermore, the very wide background distribution will result in poor detection because most of the gray level spectrum would be covered by the background model. Experiments show that



Frame 1 Frame 5 Frame 10 Frame 15

frame difference MoG our method

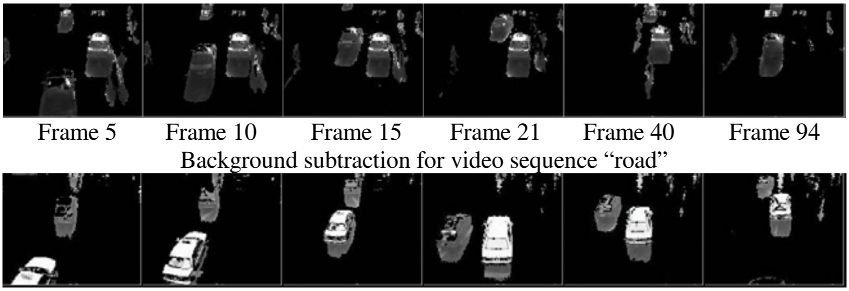
Background modeling for video sequence "road"

Frame 28 Frame 51 Frame 172 Frame 224

frame difference MoG our method

Background modeling for video sequence "road"

Fig. 3. The comparison of background modeling



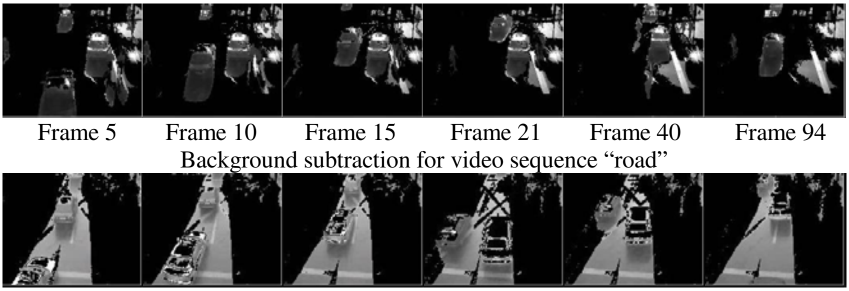
Frame 5 Frame 10 Frame 15 Frame 21 Frame 40 Frame 94

Background subtraction for video sequence "road"

Frame 51 Frame 64 Frame 81 Frame 172 Frame 186 Frame 212

Background subtraction for video sequence "car"

Fig. 4. Background subtraction by our method



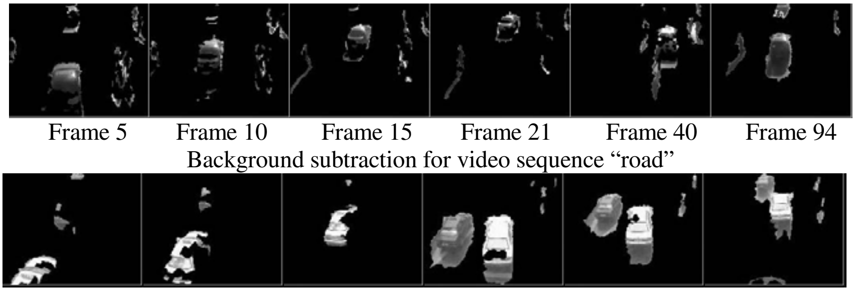
Frame 5 Frame 10 Frame 15 Frame 21 Frame 40 Frame 94

Background subtraction for video sequence "road"

Frame 51 Frame 64 Frame 81 Frame 172 Frame 186 Frame 212

Background subtraction for video sequence "car"

Fig. 5. Background subtraction by frame-difference



Frame 5 Frame 10 Frame 15 Frame 21 Frame 40 Frame 94

Background subtraction for video sequence "road"

Frame 51 Frame 64 Frame 81 Frame 172 Frame 186 Frame 212

Background subtraction for video sequence "car"

Fig. 6. Background subtraction by MoG

our method is more robust against changes in illumination. While some complicated techniques can also produce superior performance, experiments show that our simple method can produce good results with much lower computational complexity. Figure.9 shows the capability of the proposed algorithm to eliminate unwanted shadows. (a) are original frames; (b) are extracted foregrounds; (c) are foregrounds

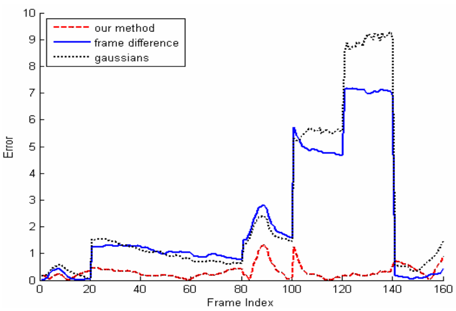


Fig. 7. Error measurement of sequence “road”

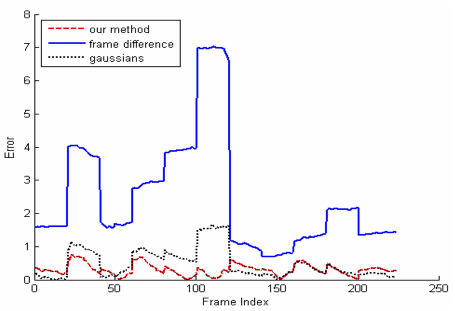
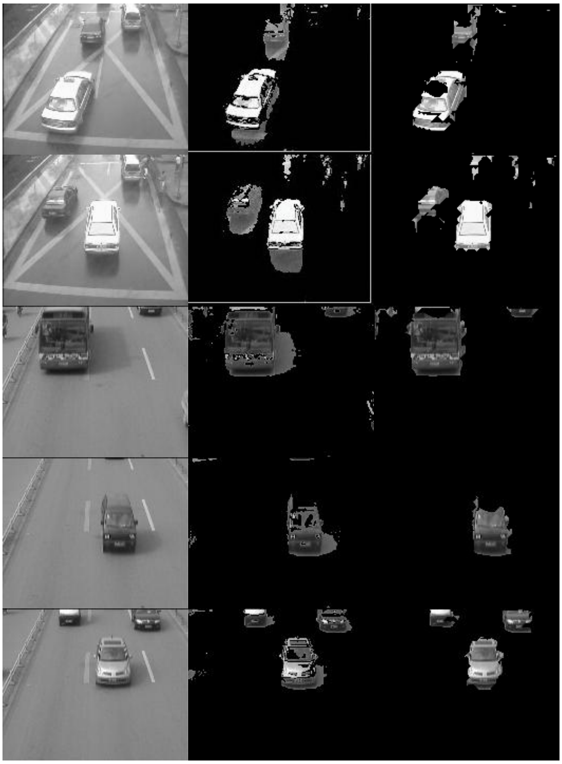


Fig. 8. Error measurement of sequence “car”



(a) (b) (c)

Fig. 9. Shadow elimination (a) original frames; (b) extracted foregrounds; (c) foregrounds after shadow elimination

after shadow elimination process. Despite multiple vehicles with different color in original image and the vehicle widows are as dark as the shadows, the shadow removal performs well and the vehicles are extracted successfully.

7 Conclusion

In this paper, we introduced a novel background model and a background subtraction technique based on wavelet theory. The model keeps a sample of intensity values for each pixel in the image and uses this sample to estimate the Marr wavelet probability density function of the pixel intensity. The density function is estimated using Marr wavelet kernel density estimation technique. Background subtraction is based on binary discrete wavelet transforms. A method is also proposed for shadow elimination from extracted foreground based on edge information. The method is robust to widely different shadow orientations, shadow appearance and foreground materials such as color, size. No matter how many shadows appear in the video sequence, all these shadows can be precisely eliminated from the background with the proposed method. Experimental results show that our method is robust against environmental noise and illumination change, and can segment the whole parts of foreground successfully.

Acknowledgments. The research is funded by Tianjin Traffic Bureau, and also funded by Tianjin Subway Safety System: [3240120], Honeywell Limited, China.

References

1. Grimson, W., Stauffer, C., Romano, R., Lee, L.: Using Adaptive Tracking to Classify and Monitor Activities in a Site. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 22–29 (1998)
2. Sen-Ching, S., Cheung, Kamath, C.: Robust Techniques for Background Subtraction in Urban Traffic Video. In: Proc. SPIE Visual Communications and Image Processing, pp. 881–892 (2004)
3. Stauffer, C., Grimson, W.E.L.: Adaptive Background Mixture Models for Real Time Tracking. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 246–252 (1999)
4. Friedman, N., Russell, S.: Image Segmentation in Video Sequences: A Probabilistic Approach. In: Proc. Thirteenth Conf. on Uncertainty in Artificial Intelligence (1997)
5. Koller, D., Weber, J., Huang, T., et al.: Towards Robust Automatic Traffic Scene Analysis in Real-time. In: Proc. Int. Conf. on Pattern Recognition, pp. 126–131 (1994)
6. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and Practice of Background Maintenance. In: Proc. IEEE Int. Conf. on Computer Vision, pp. 255–261 (1999)
7. Gupte, S., Masoud, O., Martin, R.F.K., et al.: Detection and Classification of Vehicles. IEEE Transactions on Intelligent Transportation Systems 3(1), 37–47 (2002)
8. Wang, W., Sun, M.: A New Algorithm for Detecting Object Under Traffic Scene. Computer Applications and Software 22(4), 90–92 (2005)
9. Guo, Y.T., Song, H.S., He, Y.Y.: Algorithms for Background Extraction in Video Traffic Monitoring System. Video Engineering (5), 91–93 (2006)
10. Stauer, C., Grimson, W.: Learning Patterns of Activity using Real-time Tracking. IEEE Trans.on Pattern Analysis and Machine Intelligence 22, 747–757 (2000)
11. Zhang, X.G., Ding, X.H.: A Note on Continuous Wavelet Transform. Journal of Guangxi Academy of Sciences 23(1), 4–6 (2007)

12. Swanson, M.D., Tewfik, A.H.: A Binary Wavelet Decomposition of Binary Images. IEEE Transactions on Image Processing 5(12), 1637–1650 (1996)
13. Otsu, N.: A Threshold Selection Method from Gray-Level Histogram. IEEE Trans. SMC 9(1), 62–66 (1979)
14. Elder, J., Goldberg, R.: Interactive Contour Editing. In: Proc. IEEE Conf. Computer Vision Pattern Recognition, Santa Barbara, CA, pp. 374–381 (1998)
15. Gao, T., Liu, Z.G.: Moving Video Object Segmentation based on Redundant Wavelet Transform. In: Proc. IEEE Int. Conf. on Information and Automation, pp. 156–160 (2008)
16. Gao, T., Liu, Z.G., Zhang, J.: BDWT based Moving Object Recognition and Mexico Wavelet Kernel Mean Shift Tracking. Journal of System Simulation 20(19), 5236–5239 (2008)

A Capacitated Inventory-Location Model: Formulation, Solution Approach and Preliminary Computational Results

Shu-Hsien Liao[1] and Chia-Lin Hsieh[1,2]

[1] Department of Management Sciences and Decision Making, Tamkang University,
No. 151, Yingjuan Road, Danshuei Jen, Taipei 251, Taiwan, ROC
[2] Department of Industrial Management and Enterprise Information, Aletheia University,
No. 26, Chenli Street, Danshuei Jen, Taipei 251, Taiwan, ROC
michael@mail.tku.edu.tw, hsiehcl@email.au.edu.tw

Abstract. Supply chain distribution network system provides an optimal platform for efficient and effective supply chain management. There are trade-offs between *efficiency* and *responsiveness*. In this research, a multi-objective capacitated location-inventory distribution network system is formulated which integrates the effects of facility location, transportation, and inventory issues and includes conflicting objectives. This model allows determining the optimal locations of distribution centers (DCs) and the assignment of buyers to DCs to find the set of Pareto optimal solutions. The possibility of a hybrid GA approach and its scenario analysis is investigated to understand the model performance and to illustrate how parameter changes influence its output.

Keywords: Location-inventory distribution network system, Multiobjective evolutionary algorithm, Scenario analysis.

1 Introduction

Enterprises are facing competitive environments by implementing new strategies and technologies in response to the challenges and customer demands. Recently, two generic strategies for supply chain design emerged: *efficiency* and *responsiveness*. Efficiency aims to reduce operational costs; responsiveness, on the other hand, is designed to react quickly to satisfy customer demands and save costs. In traditional distribution systems, minimizing costs or maximizing profits as a single objective is often the focus. However, very few distribution network systems are single objective problems. Multi-objective formulation has to be considered whose solutions will be a set of Pareto alternatives representing the tradeoffs among different objectives.

Recently, Daskin *et al.* [1] introduced a joint location-inventory model with risk pooling (LMRP) that incorporates inventory costs at the distribution centers (DCs) into the location problem. LMRP assumes uncapacitated DCs which is usually not practical. Capacity limitation may affect the number and locations of the facilities, the inventory that can be stored at the facilities and consequently the order frequency as

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 323–332, 2009.

well as the assignment of buyers to the facilities. Our model builds upon the initial LMRP model with some differences. First, a capacitated version of the similar model is established. Second, customer responsiveness and volume fill rate are incorporated as two extra performance metrics to make our contribution. With these considerations, we present a capacitated Multi-Objective Location-Inventory Problem (MOLIP) which results in a Mixed-Integer Non-Linear Programming (MINLP) formulation.

Evolutionary optimization algorithms are known to be efficient-solving and easy-adaptive, especially those where traditional methods failed to provide good solutions (*e.g.* MINLP). Recently, multiobjective evolutionary algorithms (MOEAs) have become prevailing since the pioneering work by Schaffer [2]. There are many efficient MOEAs that are possible to find Pareto optimal solutions as well as widely distributed sets of solutions, NSGA-II [3] is one of the most successful approaches. In our study, the well-known NSGA-II algorithm and a heuristic assignment procedure are incorporated that help to approximate the Pareto frontier for optimizing MOLIP.

This paper is organized as follows. Section 2 discusses relevant literature review. Section 3 details the model formulation. Section 4 proposes a hybrid genetic algorithm with a heuristic procedure for MOLIP. Section 5 illustrates computational results of simulated problems and considers scenario analysis to compare their performance, and finally, conclusions with some directions are provided in section 6.

2 Literature Review

Research on integrated location-inventory distribution network systems is relatively new. Nozick & Turnquist [4] proposed a joint location-inventory model to consider both cost and service responsiveness trade-offs based on the uncapacitated facility location problem. The analysis demonstrated an approximate linear safety-stock cost function in the framework and proposed a Lagrangean-based scheme. Miranda & Garrido [5] studied an MINLP model to incorporate inventory decisions into typical facility location models to solve the distribution network problem by incorporating a stochastic demand and the risk pooling phenomenon. A heuristic solution approach, based on Lagrangian relaxation and the sub-gradient method was presented. Sabri & Beamon [6] presented an integrated multi-objective multi-echelon stochastic model that simultaneously addresses strategic and operational planning decisions by developing an integrated two sub-module model. Similarly, Gaur & Ravindran [7] studied a bi-criteria optimization model to represent the inventory aggregation problem under risk pooling to find out the tradeoffs in costs and responsiveness.

Daskin *et al.* [1] and Shen *et al.* [8] present a LMRP model that incorporated safety stock placement into a location problem for a two-stage network. There are several variations of the LMRP model. Ozsen [9] presents a capacitated version of LMRP which determines the ordering policy at the DCs so that the inventory aggregation does not exceed DC capacities. A Lagrangian relaxation algorithm was applied to solve this problem. Shen & Daskin [10] extended LMRP to include the customer service component and proposed a nonlinear multi-objective model including both the cost and service objectives. They developed a weighting method and an efficient GA-based heuristic solution approach for quick and meaningful evaluation of cost/service trade-offs. From the survey, some innovative research aspects that are noteworthy have been incorporated in our research work as follows:

Multi-objective location inventory problem. Very few researches have addressed this problem. A multiobjective formulation should be required to provide the tradeoffs of Pareto optimal alternatives among total costs and customer service.

Multi-objective evolutionary algorithms (MOEAs). Most reviewed research works focused on traditional optimization techniques but few have performed successfully and efficiently. In the contrast, MOEAs have been successful developed for various optimization problems and enable the possibility for the proposed MOLIP.

3 Mathematical Formulation

3.1 Problem Description

Suppliers and distributors in general, route their products through DCs. Consider configuring a supply chain distribution network system, where a single supplier and a set of DCs are established and dispersed in a region to distribute various products to a set of buyers. The DCs act as intermediate facilities between the supplier and the buyers and facilitate the product shipment between two echelons. The supplier wishes to determine the opening DCs and to design the distribution strategy satisfying all DC capacities. Basic assumptions are used. It is assumed that all products are produced by a single supplier and one specific product for a buyer should be shipped from a single DC. Reverse flows and in-transit inventory are not considered. All demands of the buyers are uncertain. The capacities at the supplier are unlimited but capacitated at DCs. More assumptions will be stated when the mathematical model is illustrated.

Indices. i is an index set for buyers (i∈I). j is an index set of potential DCs (j∈J). k is an index set for product classifications (k∈K).

Decision Variables. Q_{wj}^{k} *is the aggregate economic order quantity for DC j for product k shipped from the supplier.* $Y_j=1$ *if DC j is open (=0, otherwise).* $X_{ji}^{k}=1$ *if DC j serves buyer i for shipping product k (=0, otherwise).*

Model Parameters. μ_j *is the capacity of DC j.* d_{ik} *is the mean demand rate for product k at buyer i.* σ_{ik} *is the standard deviation of daily demands for product k at buyer i.* ς_j^k *is the average lead time (daily) for product k to be shipped to DC j from the supplier.* ψ *is the number of days per year.* f_j *is the fixed annual facility operating cost of locating at DC site j.* h_j^k *is the annual inventory unit holding cost at DC j for product k.* o_j^k *is the ordering cost at DC j for product k per order.* tc_{ji}^k *is the unit variable transportation cost for shipping product k from DC j to buyer i.* rc_j^k *is the unit variable production and transportation cost for shipping product k from the supplier to DC j.* h_{ji} *is the distance between DC j and buyer i.* D_{wj}^k *is the expected annual demand for product k through DC j.* D_{max} *is the maximal covering distance, that is, buyers within this distance to an open DC are considered well satisfied.*

3.2 Mathematical Models

To begin modeling this problem, we also assume that the daily demand for product k at each buyer i is independent and normally distributed, *i.e.* $N(d_{ik},\sigma_{ik})$. Furthermore, at any site of DC j, we assume a continuous review inventory policy (Q_j, r_j) to meet a stochastic demand pattern. Also, we consider that the supplier takes an average lead time ς_j^k (in days) for shipping product k from the supplier to DC j so as to fulfill an order. From Eppen's inventory theory [11] considering the centralized inventory system, we assume that if the demands at different buyers are uncorrelated, the aggregate safety stock of the product k pooled at the DC j during lead time ς_j^k is normally distributed. Then, the total amount of safety stock for product k at any DC j with risk pooling is $z_{1-\alpha}\sqrt{\varsigma_j^k \sum_{i\in I}\sigma_{ik}^2 X_{ji}}$ where 1-α is referred to the *level of service* and $z_{1-\alpha}$ is the standard normal value with $P(z \le z_{1-\alpha}) = 1-\alpha$.

In our proposed model, the total cost can be decomposed into the following items: (i) *facility cost*, which is the cost of setting up DCs, (ii) *transportation cost*, which is the cost of transporting products from the supplier to the buyers via specific DCs, (iii) *operating cost*, which is the cost of running DCs, (iv) *cycle stock cost*, which is the cost of maintaining working inventory at DCs, and (v) *safety stock cost*, which is the cost of holding sufficient inventory at DCs in order to provide specific service level to their buyers. Hence, it can be represented as total cost function Z_1 as follows.

$$Z_1 = \sum_{j\in J} f_j \cdot Y_j + \psi\sum_{k\in K}\sum_{j\in J}\sum_{i\in I}(rc_j^k + tc_{ji}^k)\cdot d_{ik}\cdot X_{ji}^k + \sum_{k\in K}\cdot\sum_{j\in J} o_j^k \cdot \frac{\psi \cdot d_{wj}^k}{Q_{wj}^k}$$
$$+\sum_{k\in K}\sum_{j\in J} h_j^k \cdot \frac{Q_{wj}^k}{2}\cdot Y_j^k + \sum_{k\in K}\sum_{j\in J} h_j^k \cdot z_{1-\alpha}\sqrt{\zeta_j^k \sum_{i\in I}\sigma_{ik}^2 \cdot X_{ji}^k} \tag{1}$$

Based on Z_1, the optimal order quantity Q_{wj}^{k*} for product k at each DC j can be obtained through differentiating *eq.* (1) in terms of Q_{wj}^k, for each DC j and each product k, and equaling to zero to minimize the total supply chain cost. We can obtain $Q_{wj}^{k^*} = \sqrt{2\cdot o_j^k \cdot D_{wj}^k / h_j^k}$ for ∀ open DC j, ∀ k. In this case, there is not any capacity constraint for the order quantities Q_{wj}^k since we assume the storage capacity at the supplier is unlimited. Thus, replacing Q_{wj}^{k*} in the third and fourth terms of Z_1 in *eq.* (1), we can obtain a non-linear cost function of Z_1. In the following, we propose an innovative mathematical model for the Multi-Objective Location-Inventory Problem (MOLIP).

$$\text{Min } Z_1 = \sum_{k\in K}\sum_{j\in J} f_j \cdot Y_j^k + \sum_{k\in K}\sum_{j\in J}\sum_{i\in I}\Psi_{ji}^k \cdot X_{ji}^k + \sum_{k\in K}\cdot\sum_{j\in J}\left(\Gamma_j^k\sqrt{\sum_{i\in I} D_{ik}\cdot X_{ji}^k}\right)$$
$$+\sum_{k\in K}\sum_{j\in J} h_j^k \cdot \sqrt{\sum_{i\in I}\Lambda_{ji}^k \cdot X_{ji}^k} \tag{2}$$

$$\text{Max } Z_2 = \left(\sum_{k\in K}\sum_{i\in I} d_{ik} X_{ji}^k\right)\Big/\sum_{k\in K}\sum_{i\in I} d_{ik} \tag{3}$$

$$\text{Max } Z_3 = \left(\sum_{k\in K} \sum_{i\in I} d_{ik} \sum_{j\in \tau_i} X_{ji}^{k} \right) \Big/ \sum_{k\in K} \sum_{i\in I} d_{ik} X_{ji}^{k} \tag{4}$$

$$\text{s.t.} \quad \sum_{j\in J} \mathrm{X}_{\mathrm{ji}}^{\mathrm{k}} \le 1 \qquad \forall i\in I \; ; \forall k\in K \tag{5}$$

$$\mathrm{X}_{\mathrm{ji}}^{\mathrm{k}} \le Y_j \qquad \forall i\in I \; ; \; \forall j\in J \; ; \forall k\in K \tag{6}$$

$$\sum_{k\in K} \sum_{i\in \mathrm{I}} d_{ik} \cdot X_{ji}^{k} + \sum_{k\in K} \sqrt{\sum_{i\in I} \Lambda_{ji}^{k} \cdot X_{ji}^{k}} \le u_j \mathrm{Y}_j \quad \forall j\in J \tag{7}$$

$$X_{ji}^{k} \in \{0,1\}, \; \mathrm{Y}_j \in \{0,1\}, \quad \forall i\in I \; ; \forall j\in J \; ; \forall k\in K \tag{8}$$

where $\Psi_{ji}^{k} = \psi \cdot (rc_j^{k} + tc_{ji}^{k}) \cdot d_{ik}$ $\qquad \Gamma_j^k = \sqrt{2 \cdot \mathrm{o}_{\mathrm{j}}^{\mathrm{k}} \cdot \mathrm{h}_{\mathrm{j}}^{\mathrm{k}}}$

$D_{ik} = \psi \cdot d_{ik}$ $\qquad \Lambda_{ji}^{k} = (z_{1-\alpha})^2 \cdot \zeta_j^k \cdot \sigma_{ik}^2$

Eqs. (2)-(4) gives the objectives. While *eq.* (2) of Z_1 is to minimize the total cost, *eq.* (3) of Z_2 and *eq.* (4) of Z_3 give the objectives referred to maximizing customer service by two performance measurements: (i) *volume fill rate* (VFR), defined as the satisfied fraction of total demands without shortage; (ii) *responsiveness level* (RL), the percentage of fulfilled demand volume within specified coverage distance $D_{\max}$. *Eq.* (5) restricts a buyer to be served by a single DC if possible. *Eq.* (6) stipulates that buyers can only be assigned to open DCs. *Eq.* (7) are the maximal capacity restrictions on the opened DCs to enable the capability of holding sufficient inventory for every product that flows through the DC, and also the part of safety stock so as to maintain the specified service level. *Eq.* (8) are binary constraints. The proposed MOLIP model would not only determine the DC locations, the assignment of buyers to DCs, but also find out endogenously both the optimal order quantities and safety-stock levels at DCs. Since two of the three objective functions (Z_1 and Z_3) are nonlinear, the formulation results in an intractable multi-objective MINLP model.

4 A Genetic Approach for MOLIP

4.1 Solution Encoding

Each solution of MOLIP is encoded in a binary string of length m = |J|, where the *j*-th position indicates if DC *j* is open (= 1) or closed (= 0). This binary encoding only considers if a given DC *j* is open or closed (variables Y_j). A solution of MOLIP also involves the assignment of buyers to open DCs (variables X_{ji}). This assignment is performed by a *greedy* heuristics used to obtain the buyer-DC assignments where the buyers are sorted in the descending order of their demand flows and assign them in the sorted order to the DC according to the following rules:

Rule 1. If the buyer i is covered (i.e., there are DCs within a coverage distance), it is assigned to the DC with sufficient capacity (if exists) which can serve it with the minimal difference between the remaining capacity of an open DC j and the demand of the buyer i through DC j. That is, a DC is tried to be assigned as full as possible.

Rule 2. If the buyer i cannot be covered or there is no successful assignment from the coverage set τ_i, it is then assigned to the DC (with sufficient capacity) that increases the total cost by the least amount, regardless of its distance to the DC if possible.

However, this assignment procedure cannot guarantee that each buyer can be assigned to a satisfiable DC due to DC's capacity limitations. Thus, the infeasible solution will degrade the supplier's volume fill rate and may cause business losses.

4.2 NAGAII-Based Genetic Algorithm for MOLIP

Nondominating Sorting GA (NSGA-II) [3] is one of the best techniques for generating Pareto frontiers in MOEAs. For each solution, one has to determine how many solutions dominate it and the set of solutions to which it dominates. Thus, it ranks all solutions to form non-dominated fronts according to a *non-dominated sorting* process to classify the chromosomes into several fronts of nondominated solutions. To allow the diversification, NSGA-II also estimates the solution density surrounding a particular solution in the population by computing a *crowding distance* operator. During selection, a *crowded-comparison* operator considering both the *non-domination rank* of an individual and its *crowding distance* is used to select the offspring without lost good solutions (*elitism*), whereas crossover and mutation operators remain as usual. We summarized the NSGAII algorithm as shown in Fig. 1.

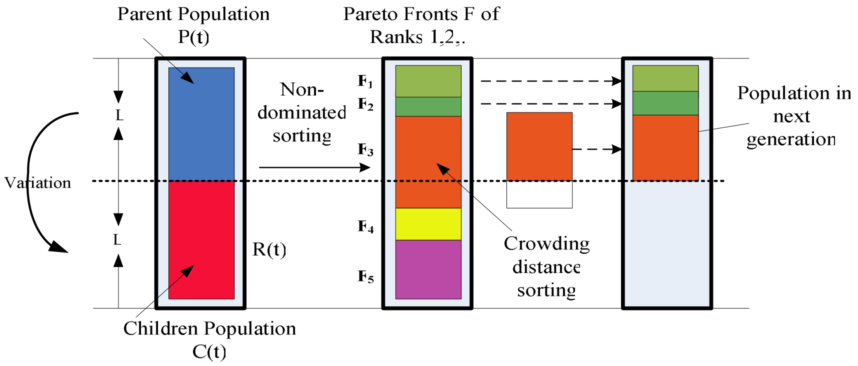


Fig. 1. Graphical Representation of the NSGA-II Algorithm

5 Computational Results and Scenario Analysis

In this section, a set of test problems similar to [12] are generated in the *base case* scenario. Then, two diverse capacity scenarios are established for scenario analysis.

5.1 Base Case Scenario (Scenario 1)

In our experiments, random sets from a square of 100 miles of width are used to generate the coordinates for DCs and buyers in all test problems. We especially establish the following parameters for the *base case* scenario. The unit transportation cost tc_{ji}

are set to 0.1 per mile. The unit production and shipping cost rc_j^k for product *k* from the supplier to DC *j* is generated uniformly on U(1,3). The average lead time ς_j^k is set to 5 days. The expected daily demands d_{ik} for product *k* at buyer *i* is generated uniformly on U(50,300). The daily inventory holding cost h_j^k at DC *j* for product *k* and the unit ordering cost o_j^k at DC *j* for product *k* are generated uniformly on U(5,10) and U(50,100), respectively. It is also assumed 365 working days ψ per year and the service level $1\text{-}\alpha$ is 0.95. The maximal covering distance is 25 miles. The capacities of DCs are set to $\mu_j = \mathrm{cap}_j \times \sum\sum \mathrm{d}_{ik}/j$, $j = 1,.,J$, $\mathrm{cap}_j \sim U(4,6)$, where the values of cap_j vary DC's capacities. Here, we want to set the DC capacity to a multiple of the mean aggregate demand of a DC. The facility costs of DCs are set to $\mathrm{f}_j = \mathrm{fac}_j \times \{U(500,1000) + U(500,1000) \times \mathrm{d}_j^{0.5}\}$, where $\mathrm{d}_j = \sum\sum \mathrm{d}_{ik}$ and the values of fac_j vary the facility operating costs. The concavity of f_j accounts for the economies of scale. In additions, the set of test problems is denoted by (*k*, *j*, *i*), where *k* is the number of products, *j* is the number of potential DCs, and *i* is the number of buyers. The values of the parameters are set to *k* = 1, 3; *j* = 5, 10, 15, 20; *i* = 30, 40, 50, 60.

We report the performance solutions of 32 problem instances for the MOLIP model in terms of objective measurements, cost components and their respective percentage of total costs. Objective measurements include the optimal solutions of *total cost* (TC), *volume fill rate* (VFR), and *responsiveness level* (RL); the cost components provide the results of *facility cost* (FC), *transportation cost* (TrC), *inventory cost* (IvC) and *safety-stock cost* (SSC) and their respective percentages of total costs (%). All objectives and cost components are expressed by the average values of their Pareto solutions and are derived by computing the average solutions of each problem instance after several iterations (say 50) for computational robustness. Since the network structure with respect to test problems may be varied when the parameters *k, j, i* are changed, we establish an index called *competitiveness level* (CL) referred to a specific DC to measure the relative capacity ratio between *supply* ability provided by a DC and *demand* requirement incurred from the distribution network system. CL indicates computational difficulty of solving capacitated problems.

In order to explore the statistical associations on 32 problem instances, the Pearson correlations among DC's capacity CL and objective measurements have been derived. It is shown that there is a significant negative correlation between CL and TC with correlation coefficient of -0.639, implying that as CL increased, the average total cost decreased. However, there is not a similar correlation between CL and VFR or RL. The correlation coefficient between these groups for VFR was -0.167, and for RL was 0.191. It is concluded that the tightly capacitated network could have overall impacts on total cost but only little effects on its volume fill rate and responsiveness level.

Next, we measure the correlations among CL and cost components which have shown significant associations for three measured variables: transportation, inventory and safety-stock costs, with correlation coefficients of -0.722, -0.550 and -0.557, respectively. However, the facility costs are the least correlated among all cost components with CL. That is, the additional capacity available in the network system enables more reduction on transportation, inventory and safety-stock costs but has no significant impact on facility costs. In depicted in Fig. 2, it is observed that those tightly capacitated problems (with small CL values) hold significantly dominating transportation costs as compared to others; however, TrC decreases considerably in those problems when capacitated environment is loosened where FC takes the place as the major cost among others. It also reveals that transportation cost is the main factor for tightly capacitated problems but facility cost dominates gradually when additional DC capacity increases.

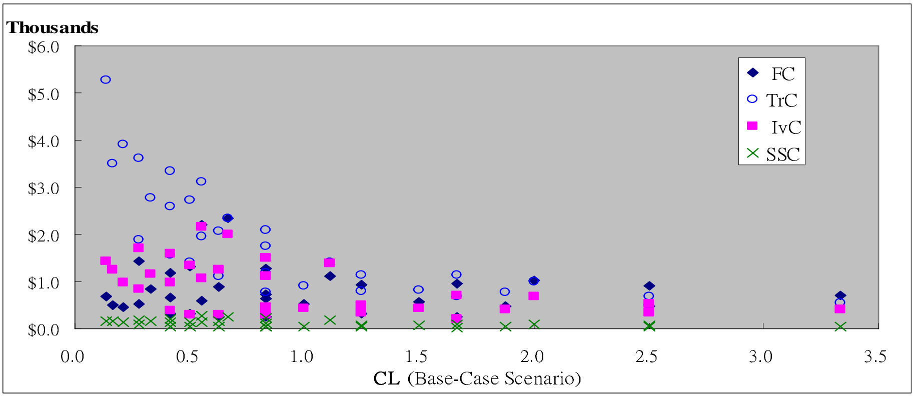


Fig. 2. Cost Components against Competitiveness Level (CL) for Scenario 1

5.2 Scenario Analysis

Our goal in this section is to illustrate the performance effects on the proposed solution procedure. We consider two diverse scenarios by changing the capacity scaling parameters of cap_j at a time as follows: cap_j ~U(2,4) for the *tight capacity* scenario (scenario 2) and cap_j~U(6,8) for *excess capacity* scenario (scenario 3). In order to see the general effects of these scenarios to the *base case scenario* (scenario 1), the incremental gap is expressed as a *percentage gap* instead of an absolute solution, which is defined as ((*value of current scenario* − *value of scenario* 1) / *value of scenario* 1) × 100%. Also, an *improvement percentage* is used to find out the relative changes of the cost component proportions. In Table 2, comparative computational results for *capacity scenarios* to the *base case scenario* are illustrated.

Table 2. Comparative Results of Capacity Scenario Analysis

Scenarios	Objectives			Percentage Gaps				Improvement Percentage			
	TC	VFR	RL	FC	TrC	IvC	SSC	%FC	%TrC	%IvC	%SSC
S_2 vs S_1	-19.3%	-21.6%	-28.8%	-22.2%	-13.9%	-21.4%	-24.9%	-1.3%	2.4%	-0.8%	-0.2%
S_3 vs S_1	6.1%	8.1%	32.8%	17.7%	0.4%	12.7%	11.81%	1.69%	-3.1%	1.27%	0.1%

The first row of S_2 vs. S_1 in Table 2 provides the comparative computational results under scenario 2 as compared to scenario 1. All the objective measurements tend to statistically decrease where TC reduced by 19.3% (in average), VFR reduced by 21.6% and RL reduced by 28.8%. As compared to scenario 1, the general effect of scenario 2 is explained that the model is simultaneously reducing the objectives due to

DC capacities shortage. The most influenced objective by DC capacity tightness is RL since the buyers could be possibly assigned to a DC that is not satisfied within the coverage distance if there are still additional capacities available. Nevertheless, if tightness causes insufficient capacities, the supplier will gradually lose his orders from buyers. That is the reason why VFR is also reduced. Contrarily, the second row of S3 vs. S1 in Table 2 provides comparative computational results under scenario 3 as compared to scenario 1, where TC increased by 6.1% (in average), VFR increased by 8.18% and RL increased by 32.8%. With sufficient capacity, the supplier can not only satisfy as many as his customers but also are capable of assigning them to nearer DCs according the *greedy* heuristic. That is why both VFR and RL are increasing at the same time and especially RL shows the largest increase of gaps among others.

The general effect on cost components is that all relevant costs are optimized in *tight capacity scenario* (scenario 2). First, FC is decreased because the optimal number of DCs is reduced. Second, TrC should have increased because it is inversely related to the number of opening DCs. The fact can be identified that only TrC has the positive improvement percentage (2.4% in Table 2) among others. However, scenario 2 reduces the buyer's willingness to place orders so as to reduce VFR. Thus, TrC is decreased for the sake of sales loss. Third, IvC is also reduced for the similar reason of the decreasing amount of opening DCs and the reduction of sales. Finally, tight capacity causes strong effects on the SSC as well. The relationship between safety stock and the number of opening DCs is explained by the square root law and the portfolio effects [13]. Tight capacity enables less risk-pooling that makes worse inventory aggregation. The larger amount of safety stock was no longer required. However, scenario 3 is contrarily different from scenario 2 that the model occurs to increase all relevant costs.

6 Concluding Remarks and Research Directions

This paper presented the MOLIP model initially represented as a multi-objective optimization formulation which examines the effects of facility location, transportation, and inventory issues. The MOLIP model via a GA approach has been successfully applied for providing promising solutions on a set of test problems and enhances the possibility of including realistic facility transportation, inventory costs in this model. The scenario analysis illustrates that excess capacity in network design is beneficial for volume fill rate and responsiveness level and has only little expense of total costs. The computational results above imply that network capacity tightness needs to be adjusted when new buyers are introduced or demand changes so as to capture the tradeoff between costs and customer service levels. The model proposed in this research is helpful in adjusting the distribution network to these changes.

An implication of this research is particularly relevant for firms seeking to increase DC's capacity flexibility in their distribution networks. To gain more benefits from their volume fill rate and responsiveness level, the firms might face the problem of increasing total cost, especially the transportation cost. However, firms could be suggested to use the services of third parties to manage down the facility costs and the transportation costs because that allows for faster facility changes in response to market or demand changes. As a result, the relative importance of facility cost in the network design declines. This proposed model can be extended in a number of ways. First, inclusion of other inventory and distribution decisions, such as inventory order policy, frequency and size of the shipments, would be a direction worth pursuing. Second, inventory at the buyers has to be explicitly modeled. Third, it is likely to include stockout and backorder costs in the model.

References

1. Daskin, M.S., Coullard, C.R., Shen, Z.M.: An inventory-location model: formulation, solution algorithm and computational results. Annals of Operations Research 110, 83–106 (2002)
2. Schaffer, J.D.: Multiple objective optimization with vector evaluated genetic algorithms. In: The First International Conference on Genetic Algorithms, Hillsdale, NJ, pp. 93–100 (1985)
3. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGAII. IEEE Trans. on Evolutionary Computation 6(2), 181–197 (2002)
4. Nozick, L.K., Turnquist, M.A.: A two-echelon allocation and distribution center location analysis. Trans. Res. Part E 37, 425–441 (2001)
5. Miranda, P.A., Garrido, R.A.: Incorporating inventory control decisions into a strategic distribution network model with stochastic demand. Trans. Res. Part E 40, 183–207 (2004)
6. Sabri, E.H., Beamon, B.M.: A multi-objective approach to simultaneous strategic and operational planning in supply chain design. Omega 28, 581–598 (2000)
7. Gaur, S., Ravindran, A.R.: A bi-criteria model for the inventory aggregation problem under risk pooling. Computers and Industrial Engineering 51, 482–501 (2006)
8. Shen, Z.J., Coullard, C.R., Daskin, M.S.: A joint location-inventory model. Transportation Science 37, 40–55 (2003)
9. Ozsen, L.: Location-inventory planning models: capacity issues and solution algorithms. PhD Dissertation, Northwestern University, Evanston, IL (2004)
10. Shen, Z.M., Daskin, M.S.: Trade-offs between customer service and cost in integrated supply chain design. Manufacturing and Service Operations Management, 188–207 (2005)
11. Eppen, G.: Effects of centralization on expected costs in a multi-location newsboy problem. Management Science 25(5), 498–501 (1979)
12. Elhedhli, S., Goffin, J.L.: Efficient Production-Distribution System Design. Management Science 51(7), 1151–1164 (2005)
13. Walter, Z., Levy, M., Bowersox, D.J.: Measuring the Effect of inventory Centralization /Decentralization on Aggregate Safety Stock. J. of Business Logistics 10(2), 1–14 (1989)

Applying Chance Discovery with Dummy Event in Technology Monitoring of Solar Cell

Tzu-Fu Chiu[1], Chao-Fu Hong[2], Ming-Yeu Wang[3], Chia-Ling Hsu[4], and Yu-Ting Chiu[5]

[1] Department of Industrial Management and Enterprise Information,
Aletheia University, Taiwan, R.O.C.
chiu@email.au.edu.tw

[2] Department of Information Management, Aletheia University, Taiwan, R.O.C.
cfhong@email.au.edu.tw

[3] Department of Bio-industry and Agribusiness Administration, National Chiayi University, Taiwan, R.O.C.
mywang@mail.ncyu.edu.tw

[4] Center for Teacher Education, Tamkang University, Taiwan, R.O.C.
clhsu@mail.tku.edu.tw

[5] Department of Information Management, National Central University, Taiwan
gloria@mgt.ncu.edu.tw

Abstract. One of the green energy, solar cell, is growing rapidly; the monitoring of its technological situation becomes critical for the stakeholders nowadays. Meanwhile, the patent data contains plentiful technological information from which is worthwhile for exploring further knowledge. Therefore, a graph-based approach, chance discovery, is employed so as to analyze the patent data, to form the technological scenarios, and to explain the overview of solar cell technology. Finally, the relationships between technology and companies, between application and companies have been observed, and the strategic suggestions have been proposed accordingly.

Keywords: Technology monitoring, chance discovery, solar cell, patent data.

1 Introduction

It is important for a company to monitor the technological situation of its own industry in order to realize the development directions of technology and to figure out the opportunities and threats within the industry. Among relevant technological information, up to 80% of all technological knowledge can be assumed to be entailed in patents [1]. Therefore, the analysis of patent data has become widespread as a tool of technology monitoring: the detection of technology trends as well as of competitive strategies, the identification both of key inventions and of key inventors, and detailed assessments of various elements of competitive strategy [2]. Apart from these depiction-based analysis, a new graph-based approach, chance discovery, will be employed in this area so as to visualize the technological overview of solar cell and to observe the relationships

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 333–342, 2009.

between technology and companies, between application and companies, as well as between technology and application.

2 Related Work

As this study attempts to monitor the technology of solar cell via chance discovery, three related works, namely technology monitoring, solar cell, and chance discovery, need to be reviewed in order to provide a foundation for the further exploration.

2.1 Technology Monitoring

Monitoring is “to watch, observe, check, and keep up with developments, usually in a well-defined area of interest for a very specific purpose”[3]. Technology monitoring (TM) is regularly searching, interpreting and disseminating relevant information on selected technology activities to provide continuing awareness of current developments and emerging trends [4]. In general, the TM process can be divided into eight steps: defining target for TM, detecting sources of information, collecting data, filtering data, analyzing data, making recommendations, disseminating and storing information, and inputting to strategy [5].

In this study, a new approach, chance discovery, will be applied in the technology monitoring area to visualize the overview of solar cell and to propose the strategic suggestions.

2.2 Solar Cell

A solar cell or photovoltaic (PV) cell is a device which converts sunlight into electricity by the photovoltaic effect [6]. Solar cell, a sort of green energy, is clean, renewable, sustainable, and good for protecting our environment. A number of materials of solar cell are currently under investigation or in mass production, including single-crystalline, poly-crystalline, amorphous (a-Si), LED (i.e., light emitting diode), TCO (i.e., transparent conductive oxide), dye-sensitized, thin-film, and compound type. In recent years (2003-2006), total PV production grew in average by almost 50% worldwide, whereas the thin-film segment grew by almost 80% (from a very low level) and reached 196 MW or 8% of total PV production in 2006 [7].

In order to understand the development situation of solar cell in U.S., this study utilizes the chance discovery approach to analyze the patent data of year 2007 from USPTO (the United States Patent and Trademark Office) [8].

2.3 Chance Discovery and Dummy Event

A chance means an event or a situation with significant impact on human decision making. Chance discovery is to become aware of a chance and to explain its significance, especially if the chance is rare and its significance is unnoticed [9]. In addition, a chance can be also conceived either as an opportunity or as a risk; where desirable effects from an opportunity should be promoted, and undesirable effects from a risk should be prevented [10].

(A) KeyGraph: In chance discovery, KeyGraph is an important technique to visualize data, which is generalized from a document-indexing method in order to extract essential events and the causal structures among them from an event sequence [11]. A KeyGraph diagram is generated from a document which is composed of sentences, and a sentence is composed of words. The algorithms for generating a KeyGraph can refer to [12].

(B) Dummy event: A dummy event, or called dark event, is neither visible nor observable [13]. Its occurrence frequency is very small or none. It diffuses randomly like an atmosphere because it does neither tend to cling to a particular event cluster nor tend to appear as a pair with a particular event. Therefore, a specific method, namely data crystallization, was proposed by Maeno and Ohsawa [13] to visualize the dummy event by applying KeyGraph. A generic data crystallization algorithm can be summarized as follows [10]: (1) event identification; (2) clustering; (3) dummy event insertion; (4) co-occurrence calculation; and (5) topology analysis. Identifying dummy events via data crystallization has been applied for designing new products of a surface inspection system (i.e., a machine for detecting defects on couple charged devices) [14].

In this study, the data crystallization algorithm will be adapted as a data adding method for including the assignee field of patent data as dummy events into the KeyGraph, so as to detect the relationships between companies and technological topics.

3 Experiment Design for Technology Monitoring

In order to monitor the technological status of solar cell, an experiment design for technology monitoring has been organized as in Fig. 1, which consists of six phases: data preprocessing, association diagram generation, topic identification, topical KeyGraph generation, dummy event insertion, and scenario for technology monitoring. The details of each phase will be depicted in the following subsections.

3.1 Data Preprocessing

In this starting phase, the patent data of solar cell (during a certain period of time) will be downloaded from the USPTO. For considering an essential part to represent a complex patent data, the abstract field is selected as the object for this study. An English POS tagger (i.e., a Part-Of-Speech tagger for English) from the University of Tokyo [15] will be employed to perform word segmenting and labeling on the patent data (i.e., the abstract field). Afterward a file of stop

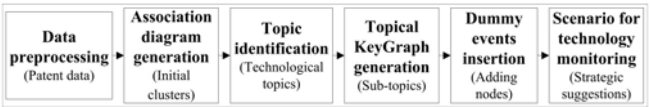


Fig. 1. Experiment design for technology monitoring

words and a file of synonyms will be built so as to facilitate the data clean-up processing. The cleaned-up patent data will be then passed onto the following phases.

3.2 Association Diagram Generation and Topic Identification

Second phase is designed to draw an association diagram using the term frequency and term association of the cleaned-up patent data. Through the proper thresholds setting of frequency and association, a number of initial clusters will be generated in the association diagram. The association is defined by the Jaccard coefficient as a measure of the co-occurrence of terms and can refer to [13].

Third phase is planned to find out the meaningful topics based on the initial clusters and domain knowledge. Among the initial clusters, domain knowledge is applied to identify the meaningful clusters by matching each initial cluster to the domain knowledge. An initial cluster will be labeled as a significant topic, only if the frequent terms of that initial cluster can correspond to the technical terms of a certain technological subfield in the domain knowledge.

3.3 Topical KeyGraph Generation

Fourth phase is designed to draw a topical KeyGraph for each significant topic correspondingly. Before drawing the KeyGraph, a group of key terms will be selected by comparing the high-frequency terms of every significant topic with the technical terms of a subfield in the domain knowledge. Using the group of key terms, a subset of patent data will be then screened from the whole data set in order to narrow down the patent data for focusing on this specific significant topic. In a topical KeyGraph, the clusters will be utilized to explain the subtopics of that significant topic and be applied for further analysis via adding dummy events afterward.

3.4 Dummy Event Insertion

Fifth phase is used to insert dummy events (i.e., nodes) into the topical KeyGraph via data crystallization, so as to explore the relationships between companies and subtopics using the assignee field of patent data. Firstly, the content of assignee field (i.e., the company name) will be added as an extra item into each patent (i.e., the abstract field). Secondly, according to this new data set, a KeyGraph will be drawn with dummy events (i.e., the company names) included. Thirdly, the associations (i.e., links) from each dummy event to other events will be recorded and be then inserted into the previous topical KeyGraphs respectively. Lastly, the co-occurrence values will be recalculated and the topical KeyGraphs will be updated subsequently.

3.5 Scenario for Technology Monitoring

The final phase of experiment design is intended to conduct the technology monitoring upon the technological status of solar cell based on the scenario derived from the dummy-event-added topical KeyGraphs. Using the relationships between subtopics and companies as well as the relevance between applications and companies, the analytical observation of solar cell technology will be executed for presenting the different aspects of technology monitoring including: the directions of technological status, relationships among technology, applications and companies, as well as strategic suggestions for companies.

4 Experiment and Explanation

The experiment has been performed according to the experiment design. The experimental results would be illustrated in the following four subsections: results of data preprocessing, results of association diagram generation and topic identification, results of topical KeyGraph generation with dummy events, and scenario of solar cell industry and technology monitoring.

4.1 Results of Data Preprocessing

As the aim of this study is to monitor the technology of solar cell, the patent documents of solar cell were the target data for this experiment. Mainly, the abstract field of patent documents was used in this study. Therefore, 81 patent items of year 2007 were collected from USPTO, using key words: “solar cell, or photovoltaic cell, or PV cell”on “title field or abstract field”. Among them, two patent documents came from Taiwan with assignees as “AU Optronics Corporation”and “Industrial Technology Research Institute”. The POS tagger was then triggered to do the data preprocessing upon the collected 81 patent items. Consequently, the patent data of year 2007 was cleaned up readily for the further processing phases.

4.2 Results of Association Diagram Generation and Topic Identification

Using the cleaned-up data from data preprocessing, the association diagram was created as in Fig. 2. In the diagram, twenty initial clusters were found while the number of comprising nodes of a cluster was set to no less than four. According to the domain knowledge, these clusters were named as: metal-oxide, nanocrystal, telemetering-system, LCD (i.e., liquid crystal display), dye-sensitized, active-semiconductor-structure, LED (i.e., light emitting diode), monitoring-terminal, burette, TCO (i.e., transparent conductive oxide), collection-region, thin-panel-enclosure, adjacent-row, thin-film-1, thin-film-2, intruder-detector, hammock, lantern, light-intensity, and spectrum from cluster ①to ⑳ respectively.

Within the 20 initial clusters, four technological topics and one application topic had been identified based on the comparison among clusters and the domain knowledge. These topics were: cluster thin-film (including ⑭ thin-film-1 and ⑮ thin-film-2), ⑤dye-sensitized, ⑦LED, ⑩ TCO, as well as cluster application (including ③telemetering-system, ⑧monitoring-terminal, ⑨burette, ⑯ intruder detector, ⑰ hammock, and ⑱ lantern). These topics were then used to explore the

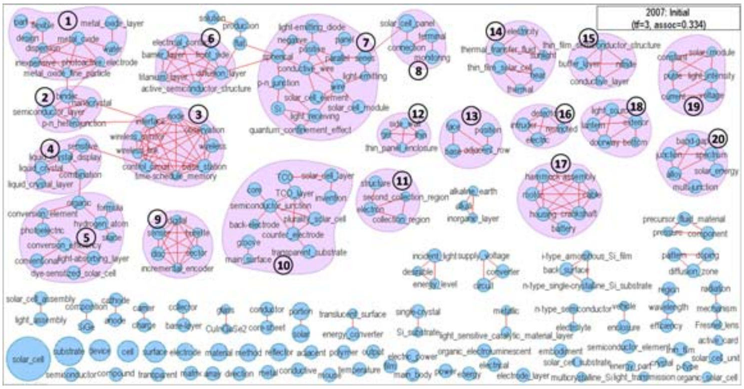


Fig. 2. Association diagram with initial clusters

subtopics within each topic, afterward the relationships between subtopics and companies, and the relationships between applications and technology.

4.3 Results of Topical KeyGraph Generation with Dummy Events

Before performing the topical KeyGraph generation, the key terms for each significant topic were necessary to be selected from that topical cluster. Using the selected key terms, a data subset was sifted for each topic from the whole patent data set. For instance, five terms, that is thin-film-solar-cell, thin-film-semiconductor-structure, thermal-transfer-fluid, nitride, and conductive-layer, were used for filtering out the irrelevant data, and the remaining data (i.e., 13 patents) were used to construct the KeyGraph of thin-film topic (in Fig. 3(a)).

After the data-subset-sifted execution, there were 13 patents for the thin-film topic, 11 patents for the dye-sensitized topic, 8 patents for the LED topic, 8 patents for the TCO topic, and 8 patents for the application topic. Fig. 3(a) to 3(e) were the KeyGraphs for five topics with their own subtopics included. In addition, all the key terms of five topics were put together to sift out a data subset for drawing an integrated KeyGraph for recognizing the relationships between applications and technological subtopics (in Fig. 3(f)).

In order to insert dummy events into the topical KeyGraph via data crystallization, the assignee field of every patent needed to be added into the abstract field and then another version of topical KeyGraph was drawn as setting the threshold of term-frequency down to one. Subsequently, the node of every company as well as its links (to other nodes) were copied and inserted into the original topical KeyGraphs respectively. These updated topical KeyGraphs were also shown in Fig. 3(a) to 3(e). Fig. 3(a), as an example, was explained in detail as follows.

In Fig. 3(a), the topical KeyGraph of thin film contained twelve subtopics with each one linking to one or several companies, which were the ①nitride-thin-film subtopic linking to LOS, the ②transparent-conductive-layer subtopic linking to

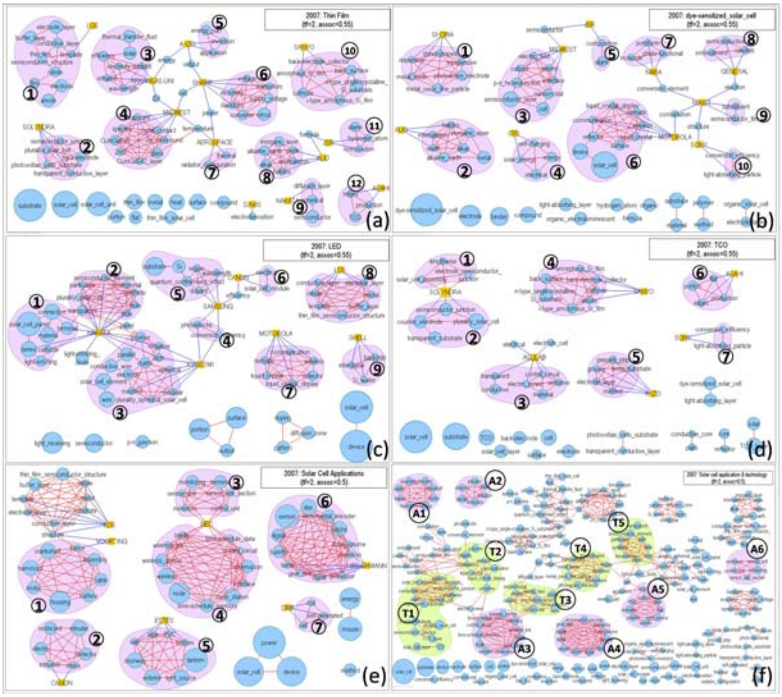

2007: Thin Film (tf=2, assoc=0.55)
(a)
2007: dye-sensitized_solar_cell (tf=2, assoc=0.55)
(b)
2007: LED (tf=2, assoc=0.55)
(c)
2007: TCO (tf=2, assoc=0.55)
(d)
2007: Solar Cell Applications (tf=2, assoc=0.5)
(e)
2007: Solar cell application & technology (tf=2, assoc=0.5)
(f)
A1
A2
A3
A4
A5
A6
T1
T2
T3
T4
T5

Fig. 3. Topical KeyGraghs with subtopics and companies

SOLYNDRA, the ③infrared-wavelength subtopic linking to ARKANSAS-UNI, the ④Zn-compound subtopic linking to MIDWEST, the ⑤energy-part subtopic linking to AKZO, and so on. Consequently, the summarized information of Fig. 3 (from 3(a) to 3(e)) was summed up in Table 1 for facilitating the following explorations.

Additionally, from the integrated KeyGraph (Fig. 3(f)), the relationships between applications and technological subtopics were not strong, only application A3 telemetering-system linking to the T1 transparent-substrate, T2 nematic-liquid-crystal, and T3 nanocrystal subtopics, and application A5 lantern linking to the T4 metal-oxide and T5 transparent-synthetic-resin subtopics. It might provide a space for applying more available technologies into various application designs.

4.4 Scenario for Technology Monitoring

According to Fig. 3 and Table 1, the scenario of solar cell industry could be observed from three aspects: technology, application, and company, and be described in their relationships between technology and companies, between applications and companies, and between applications and technology.

In the thin-film topic (Fig. 3(a)), a chain existed with five companies, namely MIDWEST, AEROSPACE, SHARP, AKZO, and ARKANSAS-UNI, linking

Table 1. Summarized information of Fig. 3 (from 3(a) to 3(e))

Fig. no.	Containing subtopics	Linking companies
3(a): thin-film	nitride-thin-film, transparent-conductive-layer, infrared-wavelength, Zn-compound, energy-part, backlight-converter, radiator-configuration, inorganic-layer, diffusion-layer, amorphous-Si-film, silane-composition, TCO-ridge	LOS, SOLYNDRA, ARKANSAS-UNI, MIDWEST, AKZO, SHARP, AEROSPACE, AUO, NAKATA, SANYO, JSR, ASAHI
3(b): dye-sensitized	metal-oxide, inorganic-layer, nanocrystal, self-charging-module, silane-composition, nematic-liquid-crystal, porphyrin-polymer, semiconductive-embodiment, semiconductive-film, light-absorbing-particle	SHOWA, AUO, MIDWEST, ITRI, JSR, MOTOROLA, NARA, GENERAL, NIAIST, SONY
3(c): LED	light-emitting-material, transparent-synthetic-resin, plurality-spherical-matrix, quantum-confinement-effect, conversion-efficiency, solar-cell-module, nematic-liquid-crystal, nitride-thin-film, interdigital- semiconductor	NAKATA, SAMSUNG, KYOSEMI, CANON, MOTOROLA, LOS, SHELL
3(d): TCO	semiconductor-junction, transparent-substrate, sensitive-material, amorphous-Si-film, temporary-substrate, flat-ridge-portion, light-absorbing-particle	SOLYNDRA, ASULAB, SANYO, AKZO, ASAHI, SONY
3(e): applications	hammock, intruder-detector, monitoring-terminal, telemetering-system, lantern, burette, computer-mouse	VOORTING, CANON, NEC, ESTES, HIRSCHMANN, DRB

together. Firstly, MIDWEST linked to the ④Zn-compound subtopic, and then linked to ARKANSAS-UNI via the Zn node, to AEROSPACE via the temperature node, and to SHARP via the cell node. Secondly, SHARP linked to the ⑥backlight-converter subtopic, and then linked to AEROSPACE via the power node, to AKZO via the output node, and to MIDWEST via the cell node. Thirdly, ARKANSAS-UNI linked to the ③infrared-wavelength subtopic, and then linked to AKZO via the energy node, and to MIDWEST via the Zn node. Apparently, these five companies had close relationships among each other in the thin-film topic. Therefore, the strategic suggestions for the individuals would be: to understand the others thoroughly as far as possible, to cooperate with the closer ones by all efforts, and to compete with the others to become a leader if strong and aggressive enough. In addition, MIDWEST and SHARP seemed stronger than others as they linked to more companies. In the dye-sensitized topic (Fig. 3(b)), a chain existed with five companies, namely NIAIST, MOTOROLA, SONY, GENERAL, and NARA, linking together. This chain had a center, NIAIST, which linked to the ⑨semiconductor-film subtopic initially, and then linked to all other companies: MOTOROLA via the combination node, SONY via the structure node, GENERAL via the electron node, and NARA via the conversion-element node. Consequently, the strategic suggestions for this special chain would be: to form a cooperative alliance through putting efforts by the central company NIAIST. Actually, NIAIST is the national institute of advanced industrial science and technology in Japan, which would be suitable to play an organizer role in a technological topic (i.e., dye-sensitized).

In the LED topic (Fig. 3(c)), a chain existed with four companies, namely NAKATA, KYOSEMI, SAMSUNG, and CANON, linking together. Firstly, NAKATA linked to the ① light-emitting-material, ② transparent-synthetic-resin, and ③ plurality-spherical-matrix subtopics, and then linked to KYOSEMI. Secondly, KYOSEMI linked to the ③ plurality-spherical-matrix and ④ conversion-efficiency subtopics. Thirdly, SAMSUNG linked to the ④ conversion-efficiency and ⑤ quantum-confinement-effect subtopics, and then linked to CANON via the efficiency node. Lastly, CANON linked to the ⑥ solar-cell-module subtopic, and then linked to SAMSUNG via the efficiency node. Thus, the strategic suggestions for the individuals would be: to stay closely with NAKATA if possible for its sufficient links (up to three subtopics), to cooperate with the closer ones by all efforts, and to compete to be a leader by NAKATA as it had a strongest link in the technological topic (i.e., LED). In fact, NAKATA represents for two well-experienced inventors (i.e., Nakata and Josuke) in Japan, who would be capable to cooperate with other companies.

In the TCO topic (Fig. 3(d)), the companies linked to subtopics individually without any links between companies. All the companies in this topic were standing alone separately. Therefore, the strategic suggestions for the individuals would be: to develop each own technological directions independently, and to find any others to cooperate with autonomously. Additionally, SOLYNDRA seemed stronger than others as it linked to the ①semiconductor-junction and ② transparent-substrate subtopics simultaneously.

In the application topic (Fig. 3(e)), the companies also linked to products individually with no links between companies. All the companies in this topic stayed alone separately. Thus, the strategic suggestions for the individuals would be: to develop each own products independently, and to find any others to cooperate with freely. Furthermore, NEC seemed sturdier than others as it linked to the ③monitoring-terminal and ④telemetering-system products at the same time.

5 Conclusion

According to the results of experiment, the meaningful topics of solar cell for year 2007 via the association diagram have been identified as follows: “thin-film”, “dye-sensitized”, “LED”, “TCO”, and “application”. These meaningful topics as well as their comprising subtopics (from topical KeyGraphs) would be helpful for the stakeholders to understand the recent situations of the entire industry. Furthermore, by exploring the relationships between technology and companies and between application and companies via adding dummy events into the KeyGraphs, the suggested strategies for some specific companies were proposed so as to provide an external viewpoint for the decision maker to facilitate his/her strategic planning. Additionally, by observing the integrated KeyGraph, the relationships between application and technology were not strong, that appeared seemingly some developing space left for applying more available technologies into various application designs.

In the future work, the patent data of solar cell can be analyzed by some other methods such as social network analysis or multivariate clustering so as to insure the validity of the experiment result. In addition, the data source can be expanded from USPTO to international scope by using the patent data of WIPO, EPO, or Japan in order to monitor the technological situation globally.

Acknowledgement: this research was supported by the National Science Council of the Republic of China under the Grants NSC 97-2410-H-415-043.

References

1. Blackman, M.: Provision of patent information: a national patent office perspective. World Patent Information 17(2), 115–123 (1995)
2. Teichert, T., Mittermayer, M.: Text mining for technology monitoring. In: Engineering Management Conference, IEMC 2002 IEEE International, vol. 2, pp. 596–601 (2002)
3. Coates, J., Coates, V., Heinz, L.: Issues management: How you can plan, organize and manage for the future. Lomond, Mt. Airy, MD (1986)
4. Ashton, W.B., Johnson, A.K., Stacey, G.S.: Monitoring science and technology for competitive advantage. In: Proceedings of 8th Annual Conference of the Society of Competitive Intelligence Professions, pp. 4–22 (1993)
5. Technology Monitoring for Business Success (1999), http://www.eirma.org/f3/cmps_index.php?page=orderpubs [2008/06/28]
6. Solar Cell, http://en.wikipedia.org/wiki/Solar_cell [2008/06/28]
7. Jager-Waldau, A.: PV status report 2007-research, solar cell production and market implementation of photovoltaics. JRC Technical Notes (2007)
8. USPTO: the United States Patent and Trademark Office, http://www.uspto.gov/ [2008/10/10]
9. Ohsawa, Y., Fukuda, H.: Chance discovery by stimulated groups of people-application to understanding consumption of rare food. Journal of Contingencies and Crisis Management 10(3), 129–138 (2002)
10. Maeno, Y., Ohsawa, Y.: Human-computer interactive annealing for discovering invisible dark events. IEEE Transactions on Industrial Electronics 54(2), 1184–1192 (2007)
11. Ohsawa, Y.: KeyGraph: visualized structure among event clusters. In: Ohsawa, Y., McBurney, P. (eds.) Chance Discovery, pp. 262–275. Springer, Heidelberg (2003)
12. Ohsawa, Y., Benson, N.E., Yachida, M.: KeyGraph: automatic indexing by cooccurrence graph based on building construction metaphor. In: Proceedings of the Advanced Digital Library Conference (IEEE ADL 1998), pp. 12–18 (1998)
13. Maeno, Y., Ohsawa, Y.: Stable Deterministic Crystallization for Discovering Hidden Hubs. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, vol. 2, pp. 1393–1398 (2006)
14. Horie, K., Maeno, Y., Ohsawa, Y.: Data crystallization applied for designing new products. Journal of Systems Science and Systems Engineering 16(1), 34–49 (2007)
15. An English POS tagger, http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/postagger/ [2008/07/28]

Color Image Retrieval Based on Interactive Genetic Algorithm

Chih-Chin Lai[1] and Ying-Chuan Chen[2]

[1] Department of Electrical Engineering
National University of Kaohsiung
Kaohsiung, Taiwan 81148
cclai@nuk.edu.tw

[2] Department of Computer Science and Information Engineering
National University of Tainan, Tainan, Taiwan 70005

Abstract. In order to efficiently and effectively retrieval the desired images from a large image database, the development of a user-friendly image retrieval system has been an important research for several decades. In this paper, we propose a content-based image retrieval method based on an interactive genetic algorithm (IGA). The mean value and the standard deviation of a color image are used as color features. In addition, we also considered the entropy based on the gray level co-occurrence matrix as the texture feature. Further, to bridge the gap between the retrieving results and the users' expectation, the IGA is employed such that the users can adjust the weight for each image according to their expectations. Experimental results are provided to illustrate the feasibility of the proposed approach.

Keywords: Content-based image retrieval, interactive genetic algorithm.

1 Introduction

With the rapid growth of the Internet, it becomes more popular to acquire and store texts, images, and graphics in digital formats. As a result, the need for content-based access digital information is becoming increasing important in multimedia information system. Analysis, classification and textual retrieval have been the target research in computational vision and digital image processing. Such methods are used in diverse areas such as entertainment, eduction, digital libraries, electrical commerce, geo-processing, crime prevention, etc.

There are two different approaches in image retrieval: text-based and content-based. The text-based approach has following drawbacks: (1) The vast amount of labor required in manual image annotation, and (2) the subjectively of human perception that causes different annotation results for a particular image. Drawbacks of the text-based image retrieval, speed up the research in the content-based image retrieval (CBIR) where the retrieval in carried out according to the image contents. There are some literatures that survey the most important CBIR systems [8,11]. Also there are some papers that overview and compare the current techniques in this area [1,12].

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 343–349, 2009.

Zheng et al. [13] proposed two new compressed-domain features for color image retrieval based on the YCbCr color space. The retrieval simulation results show that, compared with the traditional Spatial-domain Color-Histogram-based features and Vector Quantization Index Histograms-based features, the novel solutions were presented by using two kinds of features can largely improve the recall and precision performance. In recent years, the employment of feedback information to improve image retrieval precision has become a hot subject in the research field. Luo et al. [7] proposed an approach that assumes that users only need to label the images from the previous query as relevant, which generates a new vector as feedback information. Through the feature space transformation, it is an adjustment in the spatial resolution of the feature space. The spatial resolution around relevant samples is contracted. Compared with the traditional relevance feedback approach, their method is shown to obviously improve the retrieval feedback performance. Sudhamani and Venugopal [9] gave a survey that is devoted to indexing structures for objects and features. In addition, they presented an up-to-date review of various indexing structures for image retrieval.

In a CBIR system, the retrieval of images basically has been done via the similarity comparison between the query image and all candidate images in the database. To evaluate the similarity between two images, the simplest way is to calculate the distance between the feature vectors representing the two images. Color and texture are two low-level features widely used for image retrieval. In this paper, we adopt the color features, the mean value and the standard deviation, proposed by Lu and Chang [6]. In addition, we also consider the entropy [5] based on the gray level co-occurrence matrix as a texture information in an image. In order to help user find more similar or relative images, we use the IGA [10] in the CBIR system, which the system might incorporate human preference into the process of image retrieval.

The remaining of this paper is organized as follows. In Section 2, the considered image feature and IGA are briefly introduced. The proposed approach is also described in the same section. In Section 3, some experimental results are shown. Finally, conclusions are drawn in Section 4.

2 The Proposed Approach

2.1 The Color Feature

One of the key issues in CBIR is the choice of appropriate image descriptors and corresponding similarity measures. Retrieving images by color information is the most direct way among various image descriptors. Here we considered the mean values and the standard deviation of the pixel colors as the color feature of an image.

Each image in the database can be represented using three primaries of a color space. The most common color space is RGB. Thus, each pixel of a color image is represented by a vector

$$P_i = \begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix}, \tag{1}$$

where P_i is the ith pixel of the image, $1 \leq i \leq M$. R_i, G_i, and B_i are the components of primary colors red, green, blue, respectively. The M is the size of the image, and the components of P_i depict the color information.

The mean value (μ) and the standard deviation (σ) of the color image are determined as follows:

$$\mu = \frac{1}{M} \sum_{i=1}^{M} P_i \quad \text{and} \tag{2}$$

$$\sigma = \left[\frac{1}{M-1} \sum_{i=1}^{M} (P_i - \mu)^2 \right]^{1/2}, \tag{3}$$

where $\mu = [\mu_R \mu_G \mu_B]^T$ and $\sigma = [\sigma_R \sigma_G \sigma_B]^T$, each component of μ and σ indicates the RGB information, respectively.

2.2 The Texture Feature

Texture is an important image feature that has been used for characterization of images. If we can choose appropriate texture descriptor, the performance of the CBIR must be improved. In this paper, the *entropy* is used to capture texture information in an image and is defined as follows.

$$\mathit{Entropy}(E) = -\sum_i \sum_j C(i,j) \log C(i,j), \tag{4}$$

where $C(i,j)$ is the gray level co-occurrence matrix [5]. The $C(i,j)$ is obtained by first specifying a displacement vector and then counting all pairs of pixels separated by the displacement and having gray levels i and j.

2.3 Interactive Genetic Algorithm

Genetic algorithms (GAs) [3,4], within the field of evolutionary computation, are robust computational and stochastic search procedures modeled on the mechanics of natural genetic systems. GAs act as a biological metaphor and try to simulate some of the processes observed in natural evaluation. GAs are well known for their ability by efficiently exploiting the historical information to improve search performance and they have the following advantages over traditional search methods: (1) they directly work with a coding of the parameter set; (2) the search process is carried out from a population of points; (3) payoff information is used instead of derivatives or auxiliary knowledge; and (4) probabilistic transition rules are used instead of deterministic ones.

When GAs use a human to provide fitness, rather than a programmed function to compute fitness, they are called interactive GAs (IGAs). This property allows a system to be developed according to human intuition or emotion.

2.4 Image Retrieval Based on IGA

When we apply the IGA to develop a color image retrieval system, we must consider the following components: (1) a genetic representation of solutions to the problem, (2) one way to create the initial population of solutions, (3) an evaluation function that rates all candidate solutions according to their "fitness", (4) genetic operators that alter genetic composition of children during reproduction.

- Solution representation: In the proposed approach, the chromosome is made up of color feature as well as texture feature.
- Initial population: The IGA requires a population of potential solutions to be initialized at the beginning of the GA process. In our approach, we use the query results of an example image as initial candidate images. The advantage is that heuristic initialization may improve the search performance.
- Fitness function: A fitness function is the survival arbiter for individuals. Since the objective of our system is to retrieve the images that are most satisfied to the users' need, the evaluation might incorporate users' subjective evaluation and image characteristics. Thus, in our work the quality of the chromosome C in relation to the query q can be defined as Eq.(5).

$$F(q, C) = w_1(\text{similarity measure between images}) + w_2(\text{human's judgement}), \tag{5}$$

  where the coefficients w_1 and w_2 determine the relative importance of them to calculate the fitness, and $\sum w_i = 1$. The similarity measure between images is defined as

$$\text{dist}(q, C) = \sqrt{\sum_{i \in \{R,G,B\}} (\mu_i'^q - \mu_i{}'^C)^2 + \sum_{i \in \{R,G,B\}} (\sigma_i'^q - \sigma_i'^C)^2} + |E^q - E^C|, \tag{6}$$

  where $\mu_i'^q$ and $\sigma_i'^q$ represent the normalized mean value and standard deviation of the image q in i color space, respectively, and E^q is the entropy of the image q. We use a real number which is constrained to the interval $[0.0, 1.0]$ to indicate the human's judgement or preferences.
- Genetic operators: The selection operator determines which chromosomes are chosen for mating and how many offspring each selected chromosome produces. Here we adopt the tournament selection method [2] because the time complexity of it is low. The crossover operator randomly pairs chromosomes and swaps parts of their genetic information to produce new chromosomes. We use the one-point crossover in the proposed approach. In order to speedup the evaluation process, we do not use mutation operator.

3 Experimental Results

To show the effectiveness of the proposed system, some experiments will be reported. In our experiments, 500 images were chosen, consisting of 5 categories

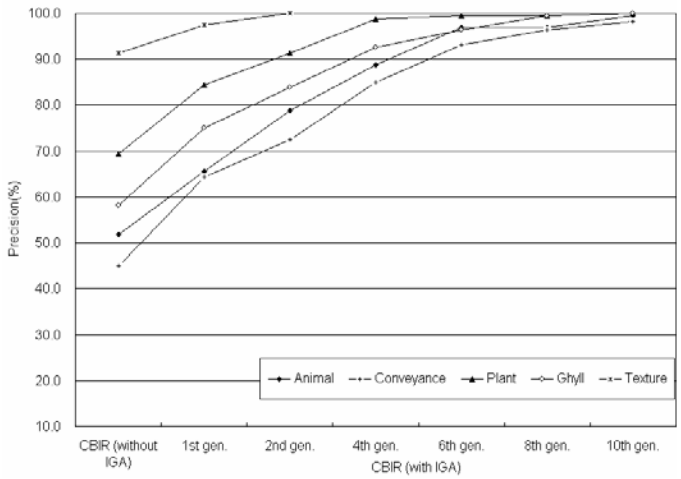


Fig. 1. The precision of the proposed approach

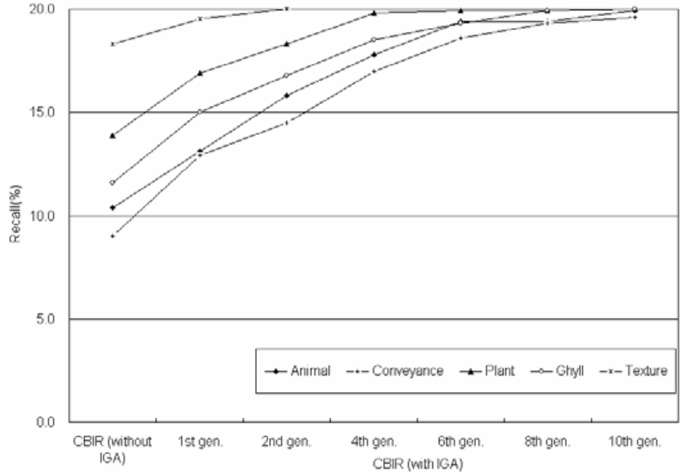
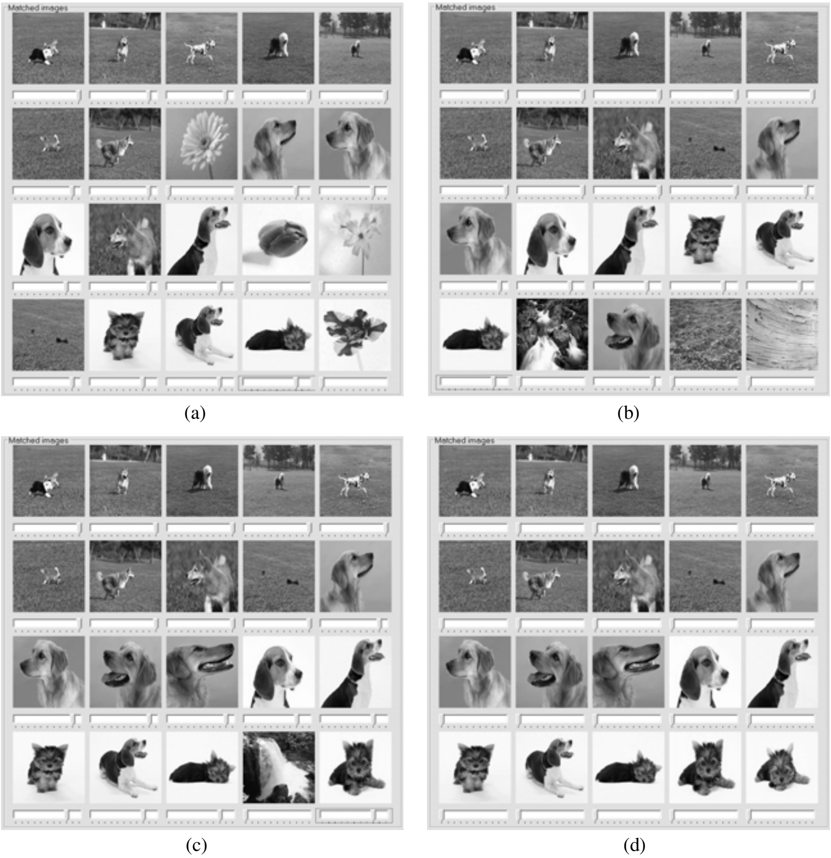


Fig. 2. The recall of the proposed approach

including animal, conveyance, plant, ghyll, and texture. The size of the images is 120 × 120 and there is no pre-processing procedure on the images.

For effectiveness, we examined how many relevant images to the query were retrieved. The retrieval effectiveness can be defined in terms of precision and recall rates. A precision rate can be defined as the fraction of retrieved images that are actually similar to the query. A recall rate is defined as the fraction of similar images in the database that are retrieved by the system. The precision and recall rates are computed by the following equations:

$$\text{precision} = \frac{N_t}{N_t + N_f} \tag{7}$$



(a)

(b)

(c)

(d)

Fig. 3. Retrieved results. (a) Initial population, (b) Population in the 1st generation, (c) Population in the 3rd generation, and (d) Population in the 4th generation.

$$\text{recall} = \frac{N_t}{N_t + N_m} \tag{8}$$

where N_t denotes the number of retrieved images similar to the query, N_f indicates the number of retrieved images dissimilar to the query, and N_m is the number of images in the database that are similar to the query but not retrieved. Here we used the top-20 retrieved images to compute the precision and recall. Figure 1 and Figure 2 show the results. According to the results, the proposed approach indeed finds more similar images with the number of generation in IGA increased.

Here we give an example to illustrate the retrieval process. An image containing a dog is used as the query image. The retrieval results are ranked in the ascending order of the fitness to the query image from the left to the right, and then from the top to the bottom. Figure 3 shows the retrieved results. From the results we find that there are more images which suit user's subjective expectation in very few generations than those in the beginning.

4 Conclusion

In this paper, an approach for a general color image retrieval was presented. The color distributions, the mean value and the standard deviation, are used as color information of an image. Additionally, texture also helps to characterize the images. In particular, we use the IGA to tune human judgment results on similarity of images. Experimental results have shown the good performance of the proposed approach. Further work of considering more low-level image descriptors (e.g., color, texture, shape, etc.) or high-level semantics in the proposed approach is in progress.

Acknowledgments. This work is partially supported by the National Science Council, Taiwan, under grant NSC 96-2221-E-390-034-MY2.

References

1. Antani, S., Kasturi, R., Jain, R.: A Survey of The Use of Pattern Recognition Methods for Abstraction, Indexing and Retrieval. Pattern Recognition 1, 945–965 (2002)
2. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Ltd., Chichester (2001)
3. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
4. Holland, J.: Adaptation in Natural and Artificial System. The University of Michigan Press, MI (1975)
5. Haralick, R.M., Shapiro, L.G.: Computer and Robot Vision, vol. I. Addison Wesley, Reading (1992)
6. Lu, T.-C., Chang, C.-C.: Color Image Retrieval Technique Based on Color Features and Image Bitmap. Information Processing and Management 43, 461–472 (2007)
7. Luo, X., Shishibori, M., Ren, F., Kita, K.: Incorporate Feature Space Transformation to Content-Based Image Retrieval with Relevance Feedback. International Journal of Innovative Computing, Information and Control 3, 1237–1250 (2007)
8. Liu, Y., Zhang, D., Lu, G., Ma, W.-Y.: A Survey of Content-Based Image Retrieval with High-Level Semantics. Pattern Recognition 40, 262–282 (2007)
9. Sudhamani, M.V., Venugopal, C.R.: Multidimensional Indexing Structures for Content-Based Image Retrieval: A Survey. International Journal of Innovative Computing, Information and Control 4, 867–881 (2008)
10. Takagi, H.: Interactive Evolutionary Computation: Cooperation of Computational Intelligence and Human Kansei. In: 5th International Conference on Soft Computing, pp. 41–50. World Scientific Press, Fukuoka (1998)
11. Veitkamp, R.C., Tanase, M.: Content-Based Image Retrieval Systems: A Survey. Technical report, UU-CS-2000-34, University of Utrecht (2000)
12. Zhou, X.S., Huang, T.S.: Relevance Feedback in Content-Based Image Retrieval: Some Recent Advances. Information Science 48, 124–137 (2002)
13. Zheng, W.-M., Lu, Z.-M., Burkhardt, H.: Color Image Retrieval Schemes Using Index Histograms Based on Various Spatial-Domain Vector Quantizers. International Journal of Innovative Computing, Information and Control 2, 1317–1326 (2006)

An Intelligent Tutoring System of Chinese Chess

Bo-Nian Chen[1,⋆], Jian-Yu Chen[2], Jr-Chang Chen[3], Tsan-sheng Hsu[4],
Pangfeng Liu[5], and Shun-Chin Hsu[6]

[1] Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
r92025@csie.ntu.edu.tw
[2] Department of Philosophy, National Chengchi University, Taipei, Taiwan
forstudyhard@gmail.com
[3] Department of Applied Mathematics,
Chung Yuan Christian University,
Taoyuan, Taiwan
jcchen@cycu.edu.tw
[4] Institute of Information Science, Academia Sinica, Taipei, Taiwan
tshsu@iis.sinica.edu.tw
[5] Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
pangfeng@csie.ntu.edu.tw
[6] Department of Information Management,
Chang Jung Christian University, Tainan, Taiwan
schsu@mail.cjcu.edu.tw

Abstract. Computer Chinese chess is an application of artificial intelligence. The playing strength of many Chinese chess programs is at the level of human masters or grandmasters. However, it is not easy for a human player to learn Chinese chess skills from these strong programs because their outputs are no more than moves and score values. It is necessary for a student to understand why he or she loses the game and to receive feedback after practice.

In this paper, we propose an intelligent tutoring system for learning Chinese chess. The system interacts with students by playing games with them and gives comments and suggestions to them during a game without any human intervention. After some iterations of practice, our system reports their learning achievements by analyzing their game records.

Keywords: Computer Chinese chess, Intelligent Tutoring System.

1 Introduction

Thanks to modern computer Chinese chess researches, many strong search programs have been developed. Computer Chinese chess programs today are at the level of human masters or grandmasters. However, it is still not easy for human players to learn Chinese chess from these programs. The strong programs output only moves and evaluation scores while human players need to understand concepts behind these moves.

[⋆] Corresponding author.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 350–359, 2009.

Human masters are good at evaluating positions and planning. They also have large amount of knowledge about opening game, middle game and endgame in their minds. Chunking mechanisms make masters recognize a position faster and easier than naive players [4].

Although masters have strong calculating ability, their computation ability is much weaker than machines. Hence, a good evaluation plus a good planning may be as important as the game-tree searching skill.

Human players evaluate a position by 1) discovering good strategies of the two players in the position, 2) reducing the position into a solved position, and 3) calculating risks under the position. They learn to play Chinese chess by 1)imitation, i.e., learning from examples, and 2) try and error strategy, i.e., learning by experiences.

It is important to practice while learning Chinese chess. Therefore, a personalized tutoring system is essential. A personalized tutoring system can teach the students according to their individual learning progress. In addition, the students can control their learning process and receive feedbacks. Finally, the system can measure the learning achievements of the students. [5]

In this paper, we propose a novel intelligent tutoring system that increases the efficiency of human learning. The tutoring system contains two modules: *the interactive tutoring module* and *the achievement analysis module*. Our idea for human learning is to give comments about the measurement of a move immediately when the move is made. Giving comments, suggestions and feedbacks can reduce the time needed for try-and-error learning.

The paper is organized as follows. In Section 2, we describe the main contributions in the past and point out the main goal in our Chinese chess tutoring system. In Section 3, we propose our method that constructs interactive tutoring system. In Section 4, we discuss the methods of achievement analysis system. In Section 5, we show our experimental results. Finally, We give a conclusion in Section 6.

2 Related Work

In chess, computer programs play equal quality to human champions. Most researcheres aim at how to improve the playing strength of computer programs. However, there are much less researches about tutoring system in chess. D. Gadwal developed an intelligent tutoring system using strategy graph for specific endgame in 1993 [3]. In 1996, Lazzeri S. used fuzzy logic and case-based reasoning to construct an intelligent consultant system for chess [7]. A. Sadikov (2006) constructed an automatic chess tutor that gives comments for human players in whole game [2].

In Chinese chess, J.-C Chen (2001) developed a client-server opening game database as an e-learning platform for human players to learn how to play good opening game [6]. There are many softwares that can play a whole game or only an endgame or browse thousands of game records [10]. When using common Chinese chess programs, the learners can distinguish good moves from bad ones by the variation of the evaluating scores. However, it is hard to identify which move results in advantage or disadvantage. There is few work about judging the performance of a student and giving comments or advices to the student in Chinese chess. However, in 2006, Matej Guid used average move difference to measure the quality of games played by world chess champions [8]. We both deal with quality of games, but the purpose and methods are different. We will compare our methods with Matej Guid's work in Section 4.

Our goal is to help the students to reduce the time of learning Chinese chess and improving the learning progress. Instead of constructing a specific game tutoring system, our subject is to construct an intelligent tutoring system for overall ability training.

3 Intelligent Chinese Chess Tutoring System

An intelligent tutoring system is an interactive system that students can perform exercise in the system and the system automatically provides feedbacks to the learner without any human intervention [9]. An intelligent Chinese chess tutoring system contains two modules. The first module, interactive tutoring module, can play with students and give comments and suggestions when they make good or bad moves. The second module, achievement analysis module, reports the students' learning achievements by analyzing their game records.

3.1 Interactive Tutoring Module

Our *interactive tutoring module* contains three elements: 1) GUI component, 2) AI game playing component, and 3) advisor component. In Chinese chess, expert's knowledge is represented by the moves played by the AI component and students' knowledge is generally represented by their game records. The most important component is the advisor component. It provides feedback to the students by commenting some of the moves that they have played and by advising better moves for those moves with bad outcomes. This process is important for improving the efficiency of learning because the earlier the conclusion is known, the less the trials are necessary for the learning process.

Our interactive tutoring module uses Jr-Chang's opening database as an judging tool for opening game. There are totally 171,302 games in Jr-Chang's opening database. It contains 44,868 win games, 22,785 loss games, and 103,649 draw games. The search engine for middle game evaluation in our system is the simplified version of Contemplation. The reason that we use simplified version is 1) it run faster than the real version of Contemplation, and 2) the accuracy is sufficient for our purpose.

3.2 Advisor in the Opening Game

In Matej Guid's work of judging the quality of games, they did not judge the quality of the opening moves. To complete the judgement in the opening game, we employed Jr-Chang's opening database. Our advisor component contains a commenting module in the opening game. To generate a comment to an opening move, we first need to define a suitable measurement for opening moves. A simple *winning number difference* measurement is as follows:

$$D(m) = s_w \times Win(m) + s_d \times Draw(m) - s_l \times Loss(m)$$

The function D is the winning number difference of the position m. The functions Win, $Draw$, and $Loss$ are the winning node, drawing node, and losing node under position m, respectively. The factor s_w is the benefit of winning a game; s_d is the benefit of ending in a draw; s_l is the cost of losing a game. We define $s_w = 2$, $s_d = 1$, $s_l = 1$. The function simply computes the benefit of selecting a branch from the opening database. The higher value of D indicates the move to be better. If we always select the best move, the measurement is very reliable. However, for general usage all reasonable branches is supposed to be selected with some probabilities. It is hard to find out correct reasonable branches by using winning number difference because there are no clear boundary in its formula. To make the selection more flexible, it is common to use a random selection to choose a move from the moves whose scores are higher than a threshold. This makes the random move sometimes worse than usual.

After discussing the drawback of the winning number difference measurement, we propose three assumptions about an opening database: 1) average quality of games in the opening database is good, 2) it contains most of the reasonable variations that are currently known, and 3) it may contain some bad moves, referred as *noise*. According to the three assumptions, we provide a new reliable measuring formula called *the advantagous ratio*:

$$WS(m) = s_w \times Win(m) + s_d \times Draw(m) + s_l \times Loss(m)$$

$$AR(m) = \frac{s_w \times Win(m) + s_d \times Draw(m)}{WS(m)} \times log_k(WS(m))$$

The function WS represents the weighted-sum of the number of nodes under position m. The function AR represents the advantageous ratio. It is considered dangerous to use ratio measurement in opening database because for those positions with equal scores, we prefer the position with more number of nodes. The *log* factor is used to improve the reliability of the analysis. The variable k is a threshold parameter that indicates the lower bound of the number of nodes under the current position that is considered as reliable. By using the *log* factor, moves with more child nodes will be considered as more important and thus can reduce the probability of confused by noise.

The advisor component is actually a classification algorithm. We classify the opening moves into two categories: 1) normal moves, and 2) questionable moves. Before we discuss about the algorithm, we first explain why there are no good moves in the opening move categories. Chinese chess is likely to be a fair game, i.e. results of the game under optimal moves should be a draw. If there is a good move in the opening game and makes one player win, it means that the player has a winning strategy and Chinese chess becomes an unfair game. Historically, the red side, who moves first, has a winning probability of two times that of the black side, but the probability of draws is more than that of wins and losses. This means that the first player does not have enough advantage to win a game. Actually, many popular opening in Chinese chess results in almost balance of power.

We define *questionable moves* as the move that leads to worse positions and *normal moves* otherwise. A move is a *forced move* if only one move is reasonable under current position. When the AR value of the best move and the AR value of another move to be judged are equal, the move is a forced move. Otherwise, the more the difference is, the worse the move is. As a result, the questionable move can be treated as a move that is worse than the best move and the difference is higher than a threshold. The value 1.0 is a good threshold. If the difference is higher than 1.0, it means either the degree of the number of nodes under the move to be judged is less than the best move or the move to be judged is less advantageous or even disadvantageous than the best move. Both the two cases show that the move to be judged is not a good choice. Furthermore, the AR value of the best move that is lower than 1.0 means either the winning rate plus drawing rate is too low, which means that the current position is in disadvantage, or the number of nodes under the current position is less than k. In other words, the database information is unreliable. In such cases, we use advisor in the middle game to judge the query move.

3.3 Advisor in the Middle Game

The evaluating score of a search engine shows the advantage of the two players. Score values vary up and down during the game. The increasing value implies a player has more advantage than the last position. Similarly, the decreasing value implies a player has less advantage than the last position. Students can realize the trend of the game by means of the change of the score values. However, even recording the variation of the score, it is not sufficient to know which move causes a disadvantage. Solving this problem by the students themselves is hard, but if a hint is given, the difficulty will be much lower. Here we provide the measurement of moves as hints that can help students find out the significant moves.

There are five categories in the middle game:

Excellent Move: a move with a clear advantage that is hard to think of
Good Move: a move with an advantage
Questionable Move: a move with a disadvantage
Bad Move: a move that causes a severe disadvantage
Normal Move: none of the above

Note that an excellent or a good move only appears when the player is in advantage while questionable move and bad move appears when the player makes a mistake.

The advisor algorithm for the middle game is shown in Algorithm 1. It applies two searches on both the current position and the next position after playing the move to be judged. The first search (with depth 8) is only used to retrieve an approximated value of the current state. If the move to be judged is equal to the computed best move, we call this move *an easy move*. If the judging move is not an easy move, we have to do the second search (with depth 12) and compare the two evaluated values.

```
// pos means current position
// m means the move to be judged
// the function Search needs two parameters: position and depth
(best_move, score) = Search(pos, 8);
next_pos = MakeMove(pos, m);
(best_next_move, next_score) = Search(next_pos, 12);
// compute the level of advantage
if(Advantage(next_score) > Advantage(score))
    value = EXCELLENT_MIDDLE;
else if(Advantage(next_score) = Advantage(score) and next_score - score > 100)
    value = GOOD_MIDDLE;
else if(best_next_move != m and next_score - score < -500 and next_score < -350)
    value = BAD_MIDDLE;
else if(best_next_move != m and next_score - score < -200)
    value = QUESTIONABLE_MIDDLE;
else
    value = NORMAL_MIDDLE;
```

Algorithm 1. Advisor in the middle game.

The strategy to judge a move is to retrieve both the evaluating score before the move and that after the move. If the two moves cause the advantage rank to increase and the score is better than the prediction of the search engine, it is considered as an excellent move. If the unexpected move only causes the score value to increase a little (more than 100 points, the value of one pawn) but the advantage rank does not change, it is considered as a good move. If the unexpected move causes the score to decrease more than 500 points (near the value of one rook), it is considered as a bad move. If the unexpected move causes the score to decrease more than 200 points but not too severe to be a bad move, it is considered as a questionable move. Examples of an excellent move and a bad move recognized by the advisor in the middle game are shown in Figure 1 and Figure 2, respectively.

By using categories in the middle game, students can understand exactly which move causes a good or a bad result. The student can retrieve Chinese chess playing skill and reduce the possibility of making mistakes by some repeated practice.

3.4 Achievement Analysis System

Interactive tutoring system serves as an advisor who always tells students where they make mistakes and give them suggestions. With the learning environment,

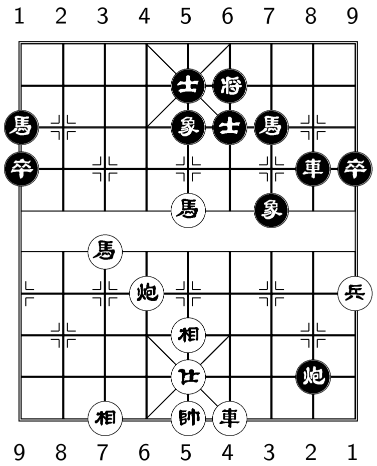


Fig. 1. An excellent move is C6=3 (played by Y.-C. Hsu)

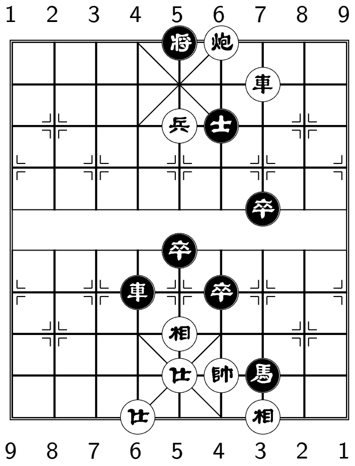


Fig. 2. A bad move is p5+1 (played by C. Chang)

students can do efficient practice of a whole Chinese chess game. After a period of learning, students may need to understand their learning achievements.

A. E. Elo's chess rating system (1978) is a widely used measurement of playing strength in chess and even Chinese chess or other games until now [1]. It is suitable for measuring the relative playing strength of a group of people. However, the ELO score does not contain direct information about the quality of the games that these players have played. Furthermore, the computation of ELO score depends on the score of other players. Therefore, it is not suitable for our purpose.

Matej Guid (2006), as described in Section 2, uses average difference between the move to be judged and the best evaluated move to compute the quality of games. They define a blunder moves as a move that losing a pawn without any compensation. They also define the complexity of positions.

Although there are many plies in a game, most moves do not clearly influence the result of the game. As a result, we only pay attention on significant moves that cause one player advantageous or disadvantageous.

In achievement analysis system, we use the categories defined in Section 3 to help our analyzing process. As described before, the possibility of the appearance of good moves and excellent moves is decided by the opponent player. Hence, the weights of the two categories should be much less than other categories. The used categories are *questionable opening moves*, *excellent middle game moves*, *good middle game moves*, *questionable middle game moves*, and *bad middle game moves*.

The analyzing process needs to perform the advisor algorithm on the whole game record. Instead of computing the average value of each category, we compute an logarithmic value of the five used categories. The formula of *the category score* is defined as follows: $CS(c) = log_{10}(n_c)/log_{10}(p)$. The function CS

represents the category score of category c. The value n_c represents the number of category c during a game. The variable p means the ply number of the game.

The idea of this formula is as follows: it is important to distinguish one good or bad move from two good or bad moves, but the importance of distinguishing ten good or bad moves from eleven good or bad moves is much less. When transforming to logarithm space, the range of CS is still $[0, 1]$, but the distribution is not linear. When the value of n_c is much less than p, the variation of CS is large; when the value of n_c is near p, the variation of CS is small. A *ranking value* is then computed by the values with a suitable weight of each category. The same idea is used to compute the ranking value. The formula of computing ranking value is as follows:

$$RV = \Sigma_c v_c \times \frac{log_{10}(\mid v_c \times CS(c) \mid +1)}{log_{10}(\mid v_c \mid +1)} - \Delta$$

The value RV is the ranking value of a player. It is a summation of weighted logarithm space of category scores among all categories. The parameter v_c is the weight of each category. Because the range of a category score is $[0, 1]$, the range of logarithm value of category score is still $[0, 1]$. Since some of the weight values are negative values, we finally subtract a value Δ that equals to the sum of all negative weight values to make sure all ranking values are positive. Note that students may play as the red side or the black side. Only the moves of one side is analyzed.

4 Experiment Results

We choose the game records of three different class of players as our test data. The first class is the 2,522 games of human masters and grandmasters. The second class is the 220 game records of a master in Xianqi Arena, whose Elo score is about 2700. The third class is the 133 game records of an expert in Xianqi Arena, whose Elo score is about 1500. All classes of games contain both red and black games for individual player. In the following reports, we define Q1 as QUESTION_OPEN, E2 as EXCELLENT_MIDDLE, G2 as GOOD_MIDDLE, Q2 as QUESTION_MIDDLE, B2 as BAD_MIDDLE. The weights of each category is shown in Table 1.

Table 1. The weight of each category

Category	Q1	E2	G2	Q2	B2
Weight	-300	200	100	-600	-1800

The time spent for the red players of Class 1 is about 128 hours; the time spent for the black players of Class 1 is about 121 hours. Class 2 and 3 can be neglected. The results of achievement analysis algorithm is shown in Table 2. The experiment computes average RV values of all players. The average RV values

Table 2. Results of achievement analysis of three different class of players

Class	Player	Games	Q1	E2	G2	Q2	B2	RV
1	Y.-C. Hsu	153	0.0143	0.0566	0.1044	0.0285	0.0026	2103.57
1	C. Lu	415	0.0238	0.0651	0.1287	0.0353	0.0046	1928.34
1	T.-H. Liu	202	0.0271	0.0740	0.1314	0.0557	0.0049	1885.53
1	T.-H. Hsu	308	0.0220	0.0558	0.1116	0.0483	0.0057	1850.54
1	J.-H. Hu	500	0.0367	0.0695	0.1375	0.0461	0.0084	1747.86
1	H.-M. Tao	133	0.0339	0.0415	0.1103	0.0539	0.0086	1717.02
1	G.-J. Chao	402	0.0268	0.0721	0.1471	0.0488	0.0112	1695.31
1	L.-C. Li	187	0.0362	0.0513	0.1202	0.0585	0.0123	1631.02
1	C. Chang	34	0.0150	0.0916	0.1487	0.0835	0.0173	1584.22
1	Y.-H. Yu	69	0.0413	0.0519	0.1300	0.0842	0.0143	1561.74
1	D.-C. Liu	75	0.0369	0.0528	0.1854	0.0913	0.0203	1480.77
1	W.-C. Yen	45	0.0437	0.0605	0.1261	0.0608	0.0210	1501.51
2	P. Jiang	220	0.0331	0.0821	0.1723	0.0511	0.0115	1695.36
3	B.-N. Chen	133	0.0636	0.0583	0.2369	0.1535	0.0393	1256.50

are computed by giving *average category scores* as parameters. The average category score is a combined value of the red games and the black games as follows:

$$\overline{CS}(c) = \frac{CS_r(c) \times n_r + CS_b(c) \times n_b}{n_r + n_b}$$

$\overline{CS}(c)$ is the average category score. $CS_r(c)$ is the category score computed by the red games; $CS_b(c)$ is the category score computed by the black games. The variable n_r means the number of red games; n_b means the number of black games.

In the result table, Y.-C. Hsu serves the best player who makes the fewest mistakes during past games. C. Lu is the second best player. The most unreliable result is the result of C. Chang because his game number (only 34) is not enough. Although it is not easy to define the reliability, we suggest that reliable data consist of more than 50 red games and 50 black games (total 100 games). To exclude C. Chang's data, the most tactical player of Class 1 is D.-H. Liu, who have the highest probability of making excellent moves. A surprising result is that T.-H. Hsu has a very high ranking value. He has a low probability of making mistakes.

P. Jiang's playing strength is about 2700 in Xianqi Arena. In our result, he plays at the level of master players. His score is even a little higher than grandmaster G.-J. Chao. However, because his opponent should be weaker than grandmasters, the probability to make mistakes may be less than master level games and the chances to make excellent moves may be more. Hence, it is true that opponent's playing strength somehow influences the RV value of the player to be judged, but the overall performance is sufficient for a student to learn and practice Chinese chess.

5 Conclusion

Today's Chinese chess program is as strong as master players. Many researches focus on increasing the playing strength of their Chinese chess programs. However, it is hard to find an intelligent system that can help improve the progress of learning Chinese chess. In this paper, we propose an intelligent tutoring system for Chinese chess that provides comments and suggestions to students. We claim that the feedback process is important for learning because it helps the student find out the key point during practice. Furthermore, by using achievement analyzer, the student can realize how much improvement that he or she has made by the practice. Experimental results show that the accuracy of the system is sufficient to give reliable comments and suggestions and it can also evaluate students' playing strength reliably.

References

1. Elo, A.E.: The rating of chess players past and present. Arco Publishing, New York (1978)
2. Sadikov, A., Mozina, M., Guid, M., Krivec, J., Bratko, I.: Automated chess tutor. In: van den Herik, H.J., Ciancarini, P., Donkers, H.H.L.M. (eds.) CG 2006. LNCS, vol. 4630, pp. 13–25. Springer, Heidelberg (2007)
3. Gadwal, D., Greer, J.E., McCalla, G.I.: Tutoring bishop-pawn endgames: an experiment in using knowledge-based chess as a domain for intelligent tutoring. Applied Intelligence 3, 207–224 (1993)
4. Gobet, F., et al.: Chunking mechanisms in human learning. Trends in Cognitive Sciences 5, 236–243 (2001)
5. Schofield, J.W., Eurich-Fulcer, R., Britt, C.L.: Teachers, computer tutors, and teaching: the artificially intelligent tutor as an agent for classroom change. American Educational Research Journal 31(3), 579–607 (1994)
6. Chen, J.C., Hsu, S.C.: Construction of online query system of opening database in computer Chinese chess. In: The 11th Conference on Artificial Intelligence and Applications (2001)
7. Lazzeri, S., Heller, R.: An intelligent consultant system for chess. Computers & Education 27(3/4), 181–196 (1996)
8. Guid, M., Bratko, I.: Computer analysis of chess champions. In: van den Herik, H.J., Ciancarini, P., Donkers, H.H.L.M(J.) (eds.) CG 2006. LNCS, vol. 4630, pp. 1–12. Springer, Heidelberg (2007)
9. Urban-Lurain, M.: Intelligent tutoring systems: an historic review in the context of the development of artificial intelligence and educational psychology (November 1998), http://www.cse.msu.edu/~urban/ITS.htm
10. Sohare information co. ltd., http://www.airgo.com.tw

Traffic Forecasts Using Interacting Multiple Model Algorithm

Yang Zhang and Yuncai Liu

Research Center of Intelligent Transportation Systems, Shanghai Jiao Tong University,
200240 Shanghai, P.R. China
{zhang-yang,whomliu}@sjtu.edu.cn

Abstract. A predictor based on interacting multiple model (IMM) algorithm is proposed to forecast hourly travel time index (TTI) data in the paper. It is the first time to propose the approach to time series prediction. Seven baseline individual predictors are selected as combination components. Experimental results demonstrate that the IMM-based predictor can significantly outperform the other predictors and provide a large improvement in stability and robustness. This reveals that the approach is practically promising in traffic forecasting.

Keywords: Traffic forecasts, Interacting multiple model (IMM), Intelligent transportation systems (ITS).

1 Introduction

The provision of accurate real-time information and short-term predictions of traffic states is fundamental to the success of intelligent transportation systems (ITS), which attracts many researchers to focus on traffic forecasting. Some papers appearing as extensive review arouse significant scientific interest in more flexible methodological approaches [1], [2]. Applying statistical methodology or heuristic methods, the empirical approaches can be approximately divided into two types: parametric and nonparametric techniques. In recent years, both techniques have shown their own advantages on different occasions, especially the latter one. They have different capabilities to capture data characteristics in linear and nonlinear domains. It seems reasonable to apply each model's unique feature to capture different patterns in the data. The complement in capturing linear and nonlinear patterns of data sets is essential for more accurate prediction. Both theoretical and empirical findings suggest that combining different methods is an effective way to improve forecast performances [3].

With three desirable properties, the interacting multiple model (IMM) algorithm has been demonstrated to be one of the most cost-effective and simple schemes for the estimation in hybrid systems [4]. It is recursive, modular and has fixed computational requirements per cycle. The importance of this approach is that it is the best compromise available currently between complexity and performance: its computational requirements are linear in the size of the problem (number of models) while its performance is almost the same as that of an algorithm with quadratic complexity.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 360–368, 2009.

Since traffic status cannot be characterized at all times by a single model, a finite number of models can adequately describe it in different regimes. Considering the characteristics of IMM algorithm and the breakthroughs of combined forecasts, we are estimated to propose a novel IMM-based predictor in the paper.

The remainder of this paper is organized as follows. First, we introduce the IMM predicting algorithm in detail. Second, we describe the data collection process and simply analyze the travel time index (TTI) data. Third, we explain the research approach, and then the results from individual and IMM-based predictor are compared in several sets of experiments with statistical analyses using figures and tables. Finally we summarize the findings of the paper.

2 Proposed Scheme

2.1 State Estimation Theory of Hybrid System

In a hybrid space $\mathbb{R}^n\times\mathbb{S}$, a discrete-time stochastic hybrid system with additive noise is defined as

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, s_{k+1}) + \boldsymbol{g}_k[s_{k+1}, \boldsymbol{x}_k, \boldsymbol{w}_k(s_{k+1}, \boldsymbol{x}_k)] \tag{1}$$

with mode-dependent noisy measurements

$$\boldsymbol{z}_k = \boldsymbol{h}_k(\boldsymbol{x}_k, s_k) + \boldsymbol{v}_k(s_k, \boldsymbol{x}_k) \tag{2}$$

where $k\in\mathbb{N}$ is a discrete-time variant; $\boldsymbol{x}_k\in\mathbb{R}^n$ is the base state vector in base state space $\mathbb{R}^n$ at time k; $s_k\in\mathbb{S}$ is the modal state (system mode index) in system mode space $\mathbb{S}$ at time k, which denotes the mode in effect during the sampling period ending at time k; $\boldsymbol{f}_k$ is the state transition function; $\boldsymbol{w}_k\in\mathbb{R}^n$ is the mode-dependent process noise with mean $\bar{\boldsymbol{w}}_k$ and covariance $\boldsymbol{Q}_k$; $\boldsymbol{g}_k$ is the noise function; $z_k\in\mathbb{R}^m$ is the measurement vector of system state at time k; $\boldsymbol{h}_k$ is the measurement function; $\boldsymbol{v}_k\in\mathbb{R}^m$ is the measurement noise with mean $\bar{\boldsymbol{v}}_k$ and covariance $\boldsymbol{R}_k$. It is assumed that $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$, uncorrelated with $\boldsymbol{x}_0$, are mutually uncorrelated. And $\boldsymbol{x}_0$ is assumed to be Gaussian with appropriate mean and covariance. Usually, $\boldsymbol{Z}^k=\{z_1, z_2, \cdots, z_k\}$ represents a set of measurement vectors and denotes the measurement sequence through time k. On the basis of the measurement sequence $\boldsymbol{Z}^k$, state prediction proposes to estimate $\boldsymbol{x}_{k+l}$ ($l > 0$), which is the main problem in the paper.

Under the assumption that the system mode sequence is a Markov chain, the Markovian transition probability of the system mode is

$$P(s_{k+1} = s^{(j)} \mid s^{(i)}, \boldsymbol{x}_k) = \phi_k(s^{(i)}, s^{(j)}, \boldsymbol{x}_k), \qquad \forall s^{(i)}, s^{(j)} \in S \tag{3}$$

where ϕ is a scalar function; $\mathbb{S}$ is the set of all modal states at all times. For multiple model (MM) estimation, a model set contains limited models $\mathcal{M}=\{m^{(j)}\}_{j=1,2,\cdots,r}$ where model $m^{(j)}$ describes the corresponding mode $s^{(i)}$ in mode space $\mathbb{S}$; r is the cardinality of the set, i.e., the possible number of system modes. Usually the former model corresponds to one or several modes, due to which the model set is smaller than the mode one. The matching relationship is shown in the formula

$$m_k^{(j)} = \{s_k = m^{(j)}\}, \; k \in \mathbb{N}, \; j = 1,2,\cdots,r \tag{4}$$

which represents that model $m^{(j)}$ is in effect at time k. Actually, model set $\mathcal{M}$ can describe system mode space $\mathbb{S}$ completely and accurately.

In general, the base state observation depends on the system mode with information in the measurement sequence. Namely, system mode sequence of the above hybrid system is a hidden Markov model (HMM) and thus, the linear stochastic hybrid system can be described as

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k(s_k)\boldsymbol{x}_k + \boldsymbol{\Gamma}_k(s_k)\boldsymbol{w}_k(s_k), \; z_k = \boldsymbol{H}_k(s_k)\boldsymbol{x}_k + \boldsymbol{v}_k(s_k) \tag{5}$$

with the mode transition governed by a first-order homogeneous Markov chain

$$P(s_{k+1} = s^{(j)} \mid s^{(i)}) = \pi_{ij}, \; \forall s^{(i)}, s^{(j)} \in S \tag{6}$$

where π_{ij} symbolizes the Markov transition probability from mode $s^{(i)}$ to mode $s^{(j)}$. The nonlinear dynamic system (jump-linear system) can be simplified as a linear one once modes of action are determinate. It is also assumed that the process and measurement noises are Gaussian mutually uncorrelated with zero mean and known covariances. The problem of hybrid state estimation is to estimate the base state and the modal state based on the measurement sequence.

2.2 IMM Predicting Algorithm

The proposed IMM predicting algorithm assumes that the modes of the system are elements of a discrete set $\mathcal{M}$ that denotes the set of all modes assumed in the MM scheme. In the process of forecasting, each predictor might be effective at time k. Its initial condition is based on the results of model filtering at prior moment k-1. In each cycle, our algorithm consists of four steps: model-conditional re-initialization, model-conditional filtering, model probability update and predicting combination. For simplicity, the paper mainly investigates two-model cases, which is fundamental to further research. The baseline IMM algorithm assumes the simple form of hybrid system described by (5) and (6).

Model-conditional Re-initialization. In this step, the input of matching filter comes from the mixed result of estimations from each filter at prior moment on condition that model j is effective in the system.

(1) Suppose that matching models are $m^{(i)}$ and $m^{(j)}$ at time k-1 and k, the mixing probability conditioned on $\boldsymbol{Z}^{k-1}$ is

$$\mu_{k-1|k-1}^{(i,j)} = P(m_{k-1}^{(i)} \mid m_k^{(j)}, \boldsymbol{Z}^{k-1}) = \pi_{ij}\mu_{k-1}^{(i)} / \bar{c}_j, \; \bar{c}_j = \sum_{i=1}^{r} \pi_{ij}\mu_{k-1}^{(i)}, \; i, j = 1,2,\cdots,r \tag{7}$$

where $\bar{c}_j$ is a normalization factor.

(2) For j = 1, 2,···, r , the mixed estimations of re-initialization state and its covariance matrix are

$$\hat{\boldsymbol{x}}_{k-1|k-1}^{(j)} = E(\boldsymbol{x}_{k-1} \mid m_k^{(j)}, \boldsymbol{Z}^{k-1}) = \sum_{i=1}^{r} \hat{\boldsymbol{x}}_{k-1|k-1}^{(i)} \mu_{k-1|k-1}^{(i,j)}, \tag{8}$$

$$\widehat{\boldsymbol{P}}_{k-1|k-1}^{(j)} = \sum_{i=1}^{r} [\boldsymbol{P}_{k-1|k-1}^{(i)} + (\hat{\boldsymbol{x}}_{k-1|k-1}^{(i)} - \hat{\hat{\boldsymbol{x}}}_{k-1|k-1}^{(j)})(\hat{\boldsymbol{x}}_{k-1|k-1}^{(i)} - \hat{\hat{\boldsymbol{x}}}_{k-1|k-1}^{(j)})^{\mathrm{T}}]\mu_{k-1|k-1}^{(i,j)} .$$

Model-conditional Filtering. Given estimation of re-initialization state and covariance matrix in the above step, model-conditional filtering updates state estimate after gaining new measurement z_k.

(1) For $i = 1, 2, \cdots, r$, the state predictions are calculated in the following:

$$\hat{\boldsymbol{x}}_{k|k-1}^{(i)} = \boldsymbol{F}_{k-1}^{(i)} \hat{\hat{\boldsymbol{x}}}_{k-1|k-1}^{(i)} + \boldsymbol{\Gamma}_{k-1}^{(i)} \overline{\boldsymbol{w}}_{k-1}^{(i)} , \quad \boldsymbol{x}_{pre(k)}^{(i)} \approx \hat{\boldsymbol{x}}_{k|k-1}^{(i)} ,$$
$$\boldsymbol{P}_{k|k-1}^{(i)} = \boldsymbol{F}_{k-1}^{(i)} \widehat{\boldsymbol{P}}_{k-1|k-1}^{(i)} (\boldsymbol{F}_{k-1}^{(i)})^{\mathrm{T}} + \boldsymbol{\Gamma}_{k-1}^{(i)} \boldsymbol{Q}_{k-1}^{(i)} (\boldsymbol{\Gamma}_{k-1}^{(i)})^{\mathrm{T}} . \tag{9}$$

The above 2nd equation denotes the predicting results at time k from predicting model i. For a discrete time series in the experiment, it is approximately equal to the state prediction.

(2) For $i = 1, 2, \cdots, r$, the measurement-prediction residual and its covariance matrix are

$$\tilde{\boldsymbol{z}}_k^{(i)} = \boldsymbol{z}_k - \boldsymbol{H}_k^{(i)} \hat{\boldsymbol{x}}_{k|k-1}^{(i)} - \overline{\boldsymbol{v}}_k^{(i)} , \quad \boldsymbol{S}_k^{(i)} = \boldsymbol{H}_k^{(i)} \boldsymbol{P}_{k|k-1}^{(i)} (\boldsymbol{H}_k^{(i)})^{\mathrm{T}} + \boldsymbol{R}_k^{(i)} . \tag{10}$$

Meanwhile, likelihood function matching model $m^{(i)}$ at time k is computed by

$$\Lambda_k^{(i)} = p(\boldsymbol{z}_k \mid m_k^{(i)}, \boldsymbol{Z}^{k-1}) \approx p[\boldsymbol{z}_k \mid m_k^{(i)}, \hat{\hat{\boldsymbol{x}}}_{k-1|k-1}^{(i)}, \boldsymbol{S}_k^{(i)}(\widehat{\boldsymbol{P}}_{k-1|k-1}^{(i)})] . \tag{11}$$

When likelihood function is assumed to be Gaussian, it can be calculated by the following formula:

$$\Lambda_k^{(i)} = p(\tilde{\boldsymbol{z}}_k^{(i)} \mid m_k^{(i)}, \boldsymbol{Z}^{k-1}) = |\, 2\pi \boldsymbol{S}_k^{(i)} |^{-1/2} \exp\{-(\tilde{\boldsymbol{z}}_k^{(i)})^{\mathrm{T}} (\boldsymbol{S}_k^{(i)})^{-1} \tilde{\boldsymbol{z}}_k^{(i)} / 2\} . \tag{12}$$

(3) For $i = 1, 2, \cdots, r$, in the step of filter update process, filter gain, state estimation update and its covariance matrix are

$$\boldsymbol{K}_k^{(i)} = \boldsymbol{P}_{k|k-1}^{(i)} (\boldsymbol{H}_k^{(i)})^{\mathrm{T}} (\boldsymbol{S}_k^{(i)})^{-1} , \quad \hat{\boldsymbol{x}}_{k|k}^{(i)} = \hat{\boldsymbol{x}}_{k|k-1}^{(i)} + \boldsymbol{K}_k^{(i)} \tilde{\boldsymbol{z}}_k^{(i)} , \quad \boldsymbol{P}_{k|k}^{(i)} = \boldsymbol{P}_{k|k-1}^{(i)} - \boldsymbol{K}_k^{(i)} \boldsymbol{S}_k^{(i)} (\boldsymbol{K}_k^{(i)})^{\mathrm{T}} . \tag{13}$$

Model Probability Update. The model probability is calculated by

$$\mu_k^{(i)} = P(m_k^{(i)} \mid \boldsymbol{Z}^k) = \Lambda_k^{(i)} \bar{c}_i / c , \quad c = \sum_{j=1}^{r} \Lambda_k^{(j)} \bar{c}_j , \quad \mu_{sup(k+1)}^{(i)} \approx \mu_k^{(i)} \tag{14}$$

where $\bar{c}_i$ is given in (7); the model probability at time k+1 is assumed to be approximately equal to that at time k for $i = 1, 2, \cdots, r$.

Predicting Combination. Giving the combination result and its covariance matrix at time k+1, the step can be described as

$$\boldsymbol{x}^{IMM} = \sum_{i=1}^{r} \boldsymbol{x}_{pre(k+1)}^{(i)} \mu_{sup(k+1)}^{(i)} ,$$
$$\boldsymbol{P}^{IMM} = \sum_{i=1}^{r} [\boldsymbol{P}_{k|k}^{(i)} + (\boldsymbol{x}_{pre(k+1)}^{IMM} - \boldsymbol{x}_{pre(k+1)}^{(i)})(\boldsymbol{x}_{pre(k+1)}^{IMM} - \boldsymbol{x}_{pre(k+1)}^{(i)})^{\mathrm{T}}]\mu_{sup(k+1)}^{(i)} . \tag{15}$$

The final prediction $\boldsymbol{x}^{IMM}$ is probability-weighted sum of forecasts from all models. And $\boldsymbol{P}^{IMM}$ is prediction error covariance matrix at time k+1. The coefficients of weight are approximately calculated from the IMM algorithm that goes into the next cycle.

3 Data Sources and Analysis

Data for this study come from the Performance Measurement System (PeMS), which can be accessed through the Internet [5]. It is a system that collects and stores freeway data from California loop detectors, and converts these data into useful information to compute freeway performance measures, which can supply different datasets as a benchmark repository. The travel time index (TTI) is commonly used in the analysis of traffic status. It expresses the average amount of extra time it takes to travel in the peak relative to free-flow travel. And it can present congestion levels in a format that is easy to understand and communicate to the general public. Considering it from a practical perspective, the PeMS supplies the TTI directly on its website to the public for reference and evaluates the traffic situation in the whole freeway network. Although TTI does not have a unit of measure, it is still a useful parameter to compare different forecasting methods efficiently.

The traffic data of 24 weeks from May 1 to Oct. 15, 2006 are used. The TTI is calculated from about 80% observed lane points in the whole transportation network of California. The 1-hour lane-aggregated TTI data are downloaded because we have access to limited traffic data. The data for a particular day start every 1 hour between 00:00 am and 23:00 pm. Fig. 1 shows the total 24 weeks traffic data hourly. The statistical autocorrelation function (ACF) helps to select the proper input dimensions of a given time series to overcome the dimensionality curse in the nonparametric approach to traffic forecasting. The ACF plots demonstrate strong seasonality and display both weekly and daily periodicity. And the ACF values reveal that our data are not as so seasonal or periodical as the others, which may lead to more difficulties in the forecasting process.

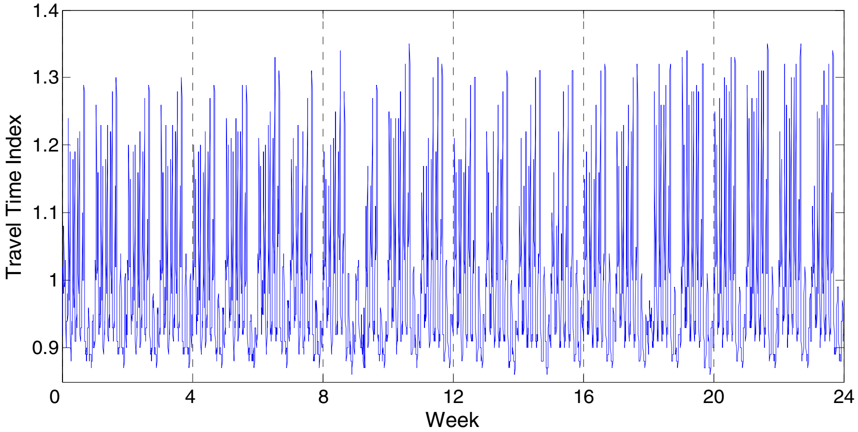


Fig. 1. Plot of all traffic data obtained from the transportation network (24 weeks)

4 Research Approach and Experiments

Seven individual predictors are selected as components: the linear least squares regression (LLSR), autoregressive moving average (ARMA), historical-mean (HM), Kalman filtering (KF), radial basis function neural network (RBF-NN), support vector regression (SVR) and least squares support vector machines (LS-SVM). They are firstly investigated as the essential preparation. Specifically, the LLSR (δ) model arises from the empirical observation that there exits a linear relationship between any $V_{\mathrm{TTI}}(t)$ and $V_{\mathrm{TTI}}(t+\delta)$ for all t and δ. The ARMA (p, q) model is one of the most frequently used parametric models in time series forecasting, which is due to their flexibility in approximating many stationary processes. K represents the number of the historical weeks used in the HM (K) model. The KF model provides an efficient algorithm for computing the one-step-ahead prediction errors and associated variances needed to produce the likelihood function. These four predictors belong to parametric techniques. For the three nonparametric techniques, we apply a state space method to determine the appropriate number of lags d ($d\in\mathbb{N}$). The RBF-NN can be greatly developed with relative ease to adapt to universal approximations with more accuracy and less time. Its outputs become linear functions of the output layer weights when the basis functions are appropriately fixed. Based on the structural risk minimization (SRM) principle, the SVR and LS-SVM models are adaptive to complex systems and robust in dealing with corrupted data. For simplicity, we choose the representative models with good performance: LLSR (1), ARMA (1, 1), HM (K=3), KF (1), RBF-NN (d=1), SVR (d=3) and LS-SVM (d=4).

For each predictor, we preserve the irregular features of the obtained TTI data in the process of forecasting. This can improve the generalization ability of the models, especially for nonparametric techniques. Due to distinct differences among the component predictors, 21 appropriately adjusted ($\boldsymbol{Q}^{(1)}$, $\boldsymbol{Q}^{(2)}$) pairs help the IMM-based predictor to obtain better performance. The IMM-based predictor provides the combination of the above forecasts. Meanwhile, the mean absolute percentage error (MAPE), variance of absolute percentage error (VAPE) and percentage error (PE) are used as measures of forecast accuracy to assess the quality of forecasts.

Table 1. Prediction performance of Situation I in MAPE & VAPE (%)

IMM	LLSR							
Predictor	MAPE	VAPE	ARMA					
LLSR	**0.9729**	**0.8704**	MAPE	VAPE	HM			
ARMA	0.9644	0.8459	**1.2267**	**1.3050**	MAPE	VAPE	KF	
HM	0.9125	0.8460	0.9798	1.2276	**1.1589**	**1.6777**	MAPE	VAPE
KF	0.9552	0.8646	1.2171	1.2910	1.0151	1.3809	**1.4472**	**1.3916**

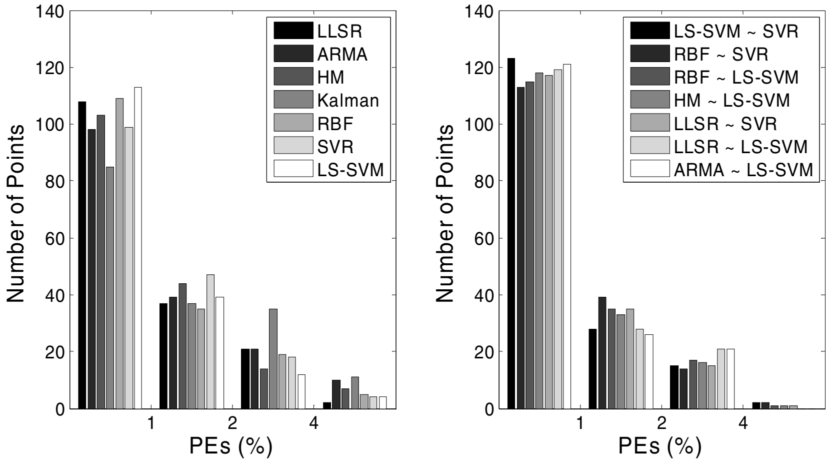
Table 2. Prediction performance of Situation II in MAPE & VAPE (%)

IMM	LLSR		ARMA		HM		KF	
Predictor	MAPE	VAPE	MAPE	VAPE	MAPE	VAPE	MAPE	VAPE
RBF-NN	0.9364	0.8674	1.0470	1.1764	0.9264	0.9475	1.0471	1.1808
SVR	*0.8443*	*0.8291*	0.8781	0.8927	0.8902	0.9668	0.9355	0.9759
LS-SVM	*0.8655*	*0.8136*	*0.8965*	*0.8326*	*0.8365*	*0.8311*	0.8867	0.8525

Table 3. Prediction performance of Situation III in MAPE & VAPE (%)

IMM Predictor	RBF-NN					
	MAPE	VAPE	SVR			
RBF-NN	**1.0629**	**1.2363**	MAPE	VAPE	LS-SVM	
SVR	*0.8583*	*0.8257*	**1.0042**	**1.0768**	MAPE	VAPE
LS-SVM	*0.8827*	*0.8437*	*0.8033*	*0.8325*	**0.9315**	**0.9773**

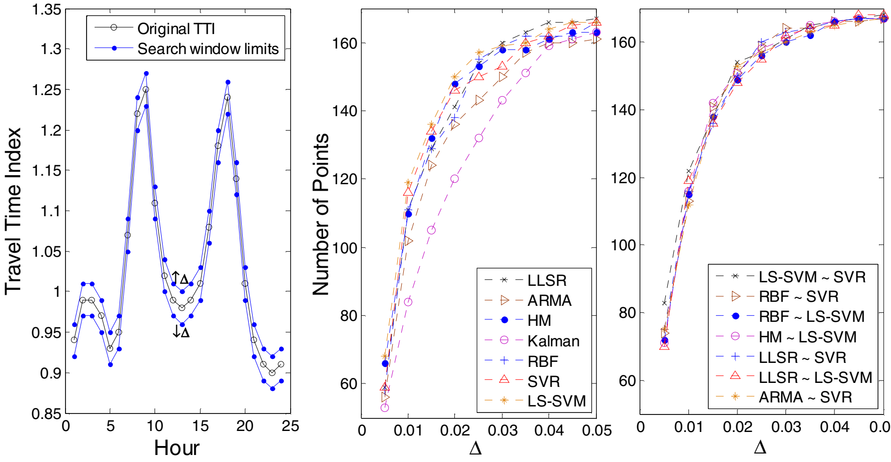
Indicated in bold, seven pairs of results from the component predictors are shown along the diagonal line of Table I and III. For example, the forecasts of ARMA and LS-SVM are shown in Table I and III with MAPE=1.2267 and MAPE=0.9315 respectively. Based on the permutations and combinations of the individual predictors in the two-model combining process, 21 categories may exist for the IMM-based model. Its results are listed below the diagonal line of each table. For instance, the combined results of HM and LLSR are shown below the diagonal of Table I with MAPE=0.9125 and VAPE=0.8460. And its counterpart of SVR and ARMA is presented with MAPE=0.8781 and VAPE=0.8927. From the tables, we can notice that all combined forecasts are better than that directly obtained from the corresponding two component predictors. To be more specific, selecting the best results of the combined forecasts (LS-SVM~SVR), we can indicate that our predictor can reduce at least 13.76% and 14.82% in MAPE and VAPE respectively compared with its individual counterpart with the best performance (LS-SVM). Moreover, compared with KF, it can reduce 44.49% and 40.18% in MAPE and VAPE respectively. These measures reflect the accuracy of the forecasting, and represent the stability of our model as well. Meanwhile, the performance of combined models can be greatly improved when the nonparametric technique is chosen as one component.



(a) Results from individual models (b) Results from IMM-based models

Fig. 2. The numbers of predicted time points lying in different ranges of |PEs|

In order to compare the IMM-based predictor with the individual ones from a statistical point of view, we choose to analyze seven combined models from the tables (indicated in italics). After calculating the PEs for each model, we statistically analyze the numbers of the predicted time points lying in different ranges of |PEs| (the absolute value of PEs). The range boundaries are set as 1%, 2%, and 4%. Fig. 2 clearly shows the comparisons in two classes. Obviously, compared with the individual models, all models in Fig. 2(b) have more predicted points lying in the range of |PEs|∈ [0, 1%] (near 120) and fewer points (less than 2) in the range of |PEs|>4%. From the figure, we can notice that the combined model with the worst performance (RBF~SVR) still performs better than the individual one with the best performance (LS-SVM). It adequately illustrates that the overall performance of the combined models can be greatly improved compared with that in Fig. 2(a).



(a) Original TTI and search window (b) Numbers of points lying in corresponding windows

Fig. 3. Analysis of forecast performance using varying search windows

Meanwhile, to produce more legible comparisons, we apply a search window method to analyze the forecast performance further. As shown in Fig. 3(a), the observed TTI $V_{\mathrm{TTI}}(t)$ (t∈ N) and a variant Δ (Δ>0, Δ∈ R) can produce the search window that is two Δ wide. At each time point t, the upper boundary of the window is $V_{\mathrm{TTI}}(t)$+Δ and the lower one is $V_{\mathrm{TTI}}(t)$-Δ. The use of the window can determine the distribution of the predicted TTI and evaluate the accuracy of each predictor in another way. Obviously, when the search window expands with Δ increasing, more predicted points lie inside it. Fig. 3(b) displays the numbers of the predicted TTI lying inside different search windows in two classes: individual and IMM-based models. As can be observed from the figure, our models have more predicted TTI falling within the window limits with Δ=0.005. Specifically, all combined models can produce over 70 points that lie inside the window. This demonstrates the effectiveness of the proposed IMM-based predictor. Moreover, the numbers of our approach increase much faster to 168 (Δ=0.05, Δ=0.055, Δ=0.055, Δ=0.065, Δ=0.055, Δ=0.045 and Δ=0.06 for LS-SVM~SVR, RBF-NN~SVR, RBF-NN~LS-SVM, HM~LS-SVM, LLSR~

SVR, LLSR~LS-SVM and ARMA~SVR models respectively) than the others (Δ=0.055, Δ=0.07, Δ=0.15, Δ=0.07, Δ=0.105, Δ=0.065 and Δ=0.08 for LLSR, ARMA, HM, KF, RBF-NN, SVR and LS-SVM models respectively). This proves the robustness of the IMM-based predictor. It can be concluded that all the IMM models can produce more accurately predicted points than the individual ones, and these points are much closer to the corresponding actual TTI. The case study also shows the effectiveness and robustness of the IMM-based predictor.

5 Conclusion

The work first proposes the IMM-based predictor and applies to time series forecasting. Hourly TTI data are used to demonstrate the effectiveness and robustness of the model, only as an example and because hourly traffic data are available to us. The predictor shows its superiority because of its extraordinary ability of compromising between complexity and performance. From a practical perspective, the remarkable adaptability evidences the potential applicability of the predictor in real-time traffic forecasting.

Acknowledgments. This research effort was supported by the National Science & Technology Supporting Program of the 11th Five-year Plan (No. 2006BAJ18B02), P. R. China.

References

1. Vlahogianni, E.I., Golias, J.C., Karlaftis, M.G.: Short-term Forecasting: Overview of Objectives and Methods. Transport Rev. 24(5), 533–557 (2004)
2. Chrobok, R., Kaumann, O., Wahle, J., Schreckenberg, M.: Different Methods of Traffic Forecast based on Real Data. Euro. J. Oper. Res. 155(3), 558–568 (2004)
3. Yu, L., Wang, S., Lai, K.K.: A Novel Nonlinear Ensemble Forecasting Model Incorporating GLAR and ANN for Foreign Exchange Rates. Comp. & Oper. Res. 32(10), 2523–2541 (2005)
4. Bar-Shalom, Y., Chang, K.C., Blom, H.A.P.: Tracking a Maneuvering Target using Input Estimation Versus the Interacting Multiple Model Algorithm. IEEE Trans. Aeros. Electron. Syst. AES-25(2), 296–300 (1989)
5. Freeway Performance Measurement System (PeMS), `http://pems.eecs.berkeley.edu`

3D Scene Analysis Using UIMA Framework

Yubin Yang[1], Wei Wei[1], Tong Lu[1], Yang Gao[1], Yao Zhang[2], and Chunsheng Yang[3]

[1] State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210093, China
[2] Jin-Ling Institute, Nanjing University, Nanjing 210093, China
[3] Institute for Information Technology, National Research Council Canada, Ottawa, Canada
yangyubin@nju.edu.cn

Abstract. With the integration of Internet and Virtual Reality techniques, three-dimensional scenes have been widely used in many fields. They are adequately vivid and complex to convey a large amount of perceptual details. However, most 3D scene data currently available are semi-structured without sufficient structured semantic clues. To address this issue, we propose a 3D scene analysis method by applying the UIMA (Unstructured Information Management Architecture) framework, which provides semantic-based intelligent annotation approaches and tools for unstructured data structures, to annotate VRML/X3D scene documents. An engine for 3D scene analysis is implemented using IBM's UIMA framework, and some experiments are carried out using the developed environment. The results demonstrated that meaningful structured features can be extracted effectively from the unstructured 3D scene data under UIMA framework.

Keywords: 3D Scene Analysis, UIMA, VRML/X3D, Semi-Structured Data.

1 Introduction

With an expansion from audio, image, and video, 3D digital models and 3D scenes are now regarded as the "4th wave of multimedia" [1]. As 3D hardware and the 3D modeling software become more and more popular, there are now a large amount of 3D models and scenes on the Web. 3D data are widely used in many areas as well, such as industrial design, electronic games, virtual reality, etc. In order to fully use the newly emerged three-dimensional resources, there is a demanding need to develop retrieval systems helping people obtain 3D scenes quickly and reuse the 3D models easily. Therefore, content-based 3D model retrieval has now become a popular research topic.

With the extensive representations of 3D models and the great increase of rich and colorful 3D scene contents, it is necessary to design a practical method to analyze 3D data and extract meaningful features of 3D scenes. Because it is a popular fact that many 3D scenes are designed by adopting virtual reality techniques (e.g. VRML/X3D, Java3D), we propose an analysis method targeting on 3D scene represented in VRML/X3D format under the UIMA (Unstructured Information Management Architecture) framework [2], which is useful to annotate the basic shape parts as features of 3D scenes.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 369–378, 2009.

The rest of this paper is organized as follows. Section 2 describes representations of 3D scenes; Section 3 introduces the UIMA framework, along with the design of a 3D scene analysis engine according to the content structures of 3D scene documents; Section 4 presents the experimental results which show the effectiveness of the annotations achieved by using UIMA-supported method in helping user retrieve satisfied semantic-based results in 3D scene analysis; finally, Section 5 concludes the paper.

2 3D Scene Representation

The current virtual environment systems use a wide variety of programming methodologies and scene description languages, many of which can also support the importation of VRML(Virtual Reality Model Language) standard. VRML is the most popular data format used for describing 3D scenes. However, with the development of Internet and the increasing amount of VRML applications, traditional VRML techniques cannot meet the demand of users any more. Therefore, Web3D Consortium published a new Internet three-dimensional standard, X3D (eXtensible 3D Specification), in 2001, in order to replace the traditional VRML standard. It is expected to provide wider scope for the development of 3D graphics on Internet.

The main goal of VRML/X3D format is to provide a precise and rich representation for 3D scenes in terms of geometric, appearance and environmental features. The data structure is a tree-like DAG (Directed-acyclic Graph) [3] shown in Fig. 1. It deploys a group of nodes to describe behaviors, interactions, geometry, and appearance of 3D scenes. The arcs reflect the manner in which the properties of each node will be altered.

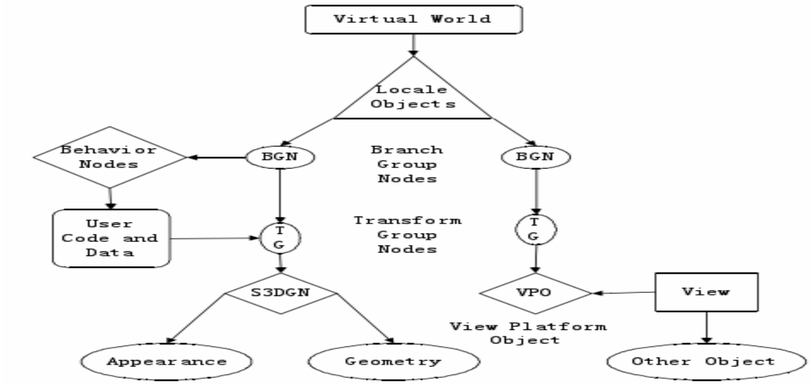


Fig. 1. The tree-like data structure of VRML/X3D scene

It can be seen that the VRML/X3D data usually contains useful information including shape, view positions, color, ambient lights, texture, textual annotation, etc. Important 3D characteristics, such as geometric primitives, geometric transformations, composite objects, alternate contents, multi-level representations, environmental elements, and metadata information, are all represented as important nodes in VRML/X3D files. However, it is also obvious that the VRML/X3D data are organized as a semi-structure form, without structured index that could help lead to semantic understanding. To address this issue, this paper proposes a method to extract useful semantic features from the semi-structured VRML/X3D documents by adopting a unified architecture to integrate the VRML/X3D data and analyze them.

3 3D Scene Analysis under UIMA Framework

3.1 UIMA Architecture and Benefits

IBM's UIMA is an open, industrial-strength, scalable and extensible platform for creating, integrating, and deploying unstructured information management solutions from powerful text or multi-modal analysis and search components [2]. At present, three-dimensional scenic tools based on VRML/X3D technologies are mostly text-based, which makes UIMA a perfect framework to analyze 3D VRML/X3D scenes.

Firstly, designing a high-performance semantic-based 3D scene analysis framework presents as a big challenge itself. The application of UIMA will enable us to devote most of our time and efforts to study 3D models themselves without focusing on system implementation details.

Secondly, as software-developing architecture capable of analyzing large volumes of unstructured information for the purpose of discovering, organizing, and delivering relevant knowledge to end users, UIMA supports key technologies that play important roles in our project, such as Knowledge Discovery, Semantic Web and Ontology [4]. This is the main reason for us to choose UIMA.

Finally, UIMA is free and open. Plenty of materials including SDK and valuable guides can be downloaded freely, which enable us to grip UIMA with ease.

3.2 Analysis Solution Based on UIMA

By using the UIMA architecture, the framework of 3D VRML/X3D scene analysis solution is illustrated in Fig. 2. It includes three components: (1) raw 3D data exchanging, (2) X3D data processing, and (3) XML data storage in relational database. Considering the large volume of information, high-level contents in the vast collections of 3D raw documents are, unfortunately, buried in lots of noises. Searching for what we need or performing sophisticated data mining over those unstructured information sources presents new challenges. The UIMA application may be generally characterized as a software system that analyzes large volume of unstructured information (text, images, audio, video, modelling file, etc.) to discover, organize, and deliver relevant knowledge to the client or application end-user.

Fig. 2 presents a semantic annotation system for 3D scene analysis under UIMA's support. An Analysis Engine (AE) is a program that analyzes 3D documents and infers information from them. An AE annotates 3D virtual scenes interactively via relevant semantic analysis mechanism. Primitive AE contains a single annotator, otherwise Aggregate AE contain multiple annotators.

An annotator is a component that contains analysis logic. Annotators analyze a 3D file and create additional data (metadata) about the file. Annotators produce their results of analysis in the form of typed Feature Structures, which are simply data structures that have a type and a set of *(attribute, value)* pairs. An annotation is a particular type of feature structure attached to a region of the analyzed 3D file (for example, a meaningful shape object in a 3D scene).

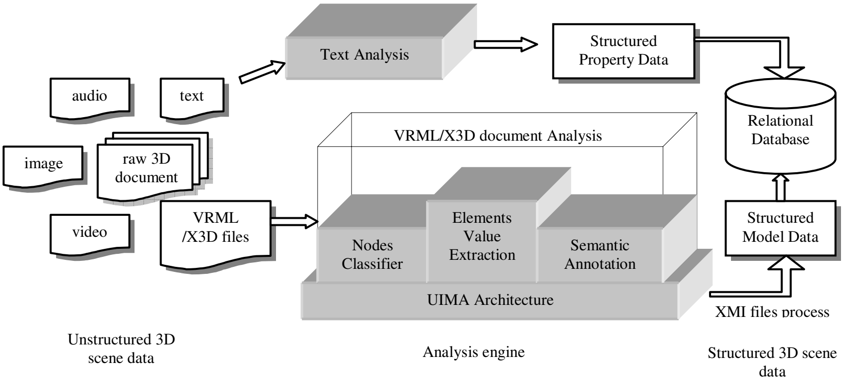


Fig. 2. Framework of 3D scene analysis

All feature structures, including annotations, are represented in the UIMA Common Analysis Structure (CAS). The CAS is the core data structure through which all UIMA components communicate each other. CAS can represent arbitrary types of feature structures, and feature structures can refer to other feature structures. Therefore, we can use CAS to represent a parse tree for a 3D scene document.

The functions of the analysis engine module will be described in detail later.

3.3 3D Scene Analysis Levels

As a new approach of semantic-based 3D data analysis via UIMA, our proposed method addresses the problem of traditional 3D feature extraction algorithms that the selected subsets of data are not representative and discriminative enough. The fundamental objectives of our research ideas are to train computers to understand 3D scene data at semantic level, in order to achieve efficient 3D model recognition and retrieval.

Whereas there are many difficulties for human being to build perception and cognition of 3D depth information and complicated relationships in a virtual 3D scene, we can replace a complicated scene with a simple geometric prototype model, which just satisfies the necessity of visual perception. This work is able to simplify a 3D scene into a group of annotations for the main objects in the scene, which helps people understand the complex scene easier.

And therefore, the analysis granularities are divided into two levels: document-level and collection-level [5]. In document-level, computers can discover knowledge from a 3D scene document that reveals relationships between the unstructured data and the structured semantic meanings attached to the data. In collection-level, a set of 3D scenes will be learned to be classified based on their structured semantic meanings.

Document-level analysis. A raw 3D scene usually does not contain any semantic information but geometric information (perhaps, appearance properties also). For the sake of simplicity, assume a 3D scene is composed of basic geometric models. Thus, geometric information in a 3D scene generally includes traditional shapes, and the scene spatial relationships in the scene are declared by position and viewpoint information in scene scripts.

Nevertheless, separate shapes are not able to present integrated semantic concepts clearly. To solve this problem, we consider that a 3D partial scene is a collection of basic models, and then objectives in this level can be depicted as detecting sub-model relationships and tagging semantic annotation to each section.

Collection-level analysis. The document-level analysis focuses on an individual 3D scene document, but in this level, the focus is on a set of 3D model files, or a series of 3D scene documents. All the available resources here include the original 3D documents, semantic description counterparts, and the index database. By using those resources as the processed elements, we would develop "3D Model Classifier" and "3D Scene Recognition" component for practical application in this level.

3.4 Analyzing Process for 3D Scene Documents

Because we regard a 3D scene modeling file as a text document (file in .wrl, or .x3d format), each shape in a 3D scene is then associated with specific keywords. Semantically-related shapes will be assembled as a so-called "section phrase" (a set of shapes which compose a spatial section), and related "section phrases" will further be assembled as a "scene sentence" (a set of sections which compose the whole scene). Accordingly, this process would repeat for several times until concepts are generated to satisfy semantic expression of a 3D scene. After the process, we are able to obtain a description document about the semantic hierarchy information of a 3D scene.

In this procedure, loose and unstructured data are gathered repeatedly according to their semantic meanings while the original 3D scene is segmented repeatedly. Therefore, this procedure of 3D scene segmentation and feature extraction could be described as a Data Set provides materials for Semantic Tagging Device to further process, with the help of Data Relationship Detector and Knowledge Database [5].

3.5 Annotation Design in Analysis Engine

The request for 3D scene data processing is always complex, particularly for 3D scenes in complicated applications like multimedia games. With the maturity of the UIMA tools and the popularity of XML techniques, more and more analysis engine of multimedia data have been integrated into unified architecture of document analysis, and structured data is thus generated and exported to relational database for information retrieval purpose.

We employ UIMA tools to implement the VRML/X3D document analysis engine. The main contribution focuses on the following three aspects: (1) Scene data structure analysis; (2) Annotation definition; (3) Analysis results storage. The modules are illustrated in Fig. 3.

Firstly, we adopt compatible VRML/X3D document format as the scene graph data structure, for this kind of scene graph can easily express the users' design intents and it is a type of semi-structure data that needs semantic feature extraction.

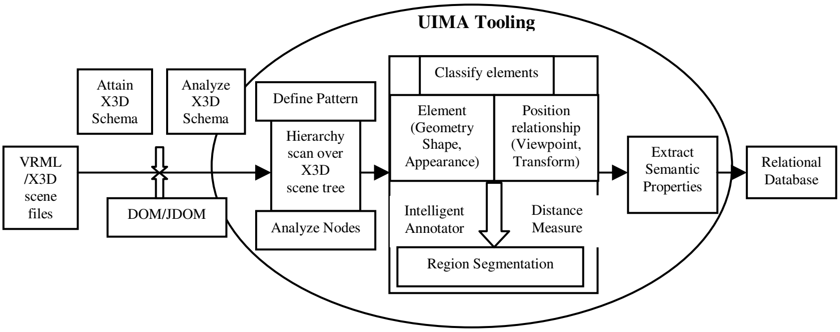


Fig. 3. Modules of VRML/X3D scene document Analysis process

Secondly, designing partial patterns is to annotate the feature nodes. In our method, some basic geometry shapes (Cube, Box, Sphere, Cylinder, Cone or IndexedFaceSet, etc.) would be annotated. Those basic prototypes are defined as regular patterns and can be recognized by corresponding annotators. Annotation algorithms based on the regular patterns will compare the current 3D geometry parts in a 3D scene with the defined basic shapes. The combination of the recognized basic geometry shapes and their values is implemented by the semantic-based analysis engine. An annotator provides the standard interfaces and possesses several functions. The main functions in an annotator are: *initialize*, *process*, and *destroy*.

(1) *Initialize* is called by the framework once it first creates an instance of the annotator class.
(2) *Process* is called once each item is processed.
(3) *Destroy* may be called by the application when it is finish using the annotator.

We design annotator class extending the JCasAnnotator_ImplBase using the JCas, which is provided in the UIMA framework. This class has implementations for all required functions except for the *process* function. So we only need to implement the *process* method for analysis engine. After defining regular expressions by some Java class fields, the basic shape feature patterns can be hold and will be used in the *process* method, which is typically called once for each document. In detail, a Matcher class is part of the java.util.regex package, and is used to find the defined pattern in the document text [2]. When a pattern is found, it is as simple as creating a new Java object to record the annotation.

Thirdly, we call functions to add the new annotation to indices maintained in CAS in UIMA framework, which is a subsystem in it for exchanging data between the various UIMA components, such as analysis engines and unstructured information management applications. By default, the CAS implementation used for analyzing text documents keeps an index of all annotations in their order from the beginning to end of the document. Subsequent annotators or applications use the indices to iterate over the annotations.

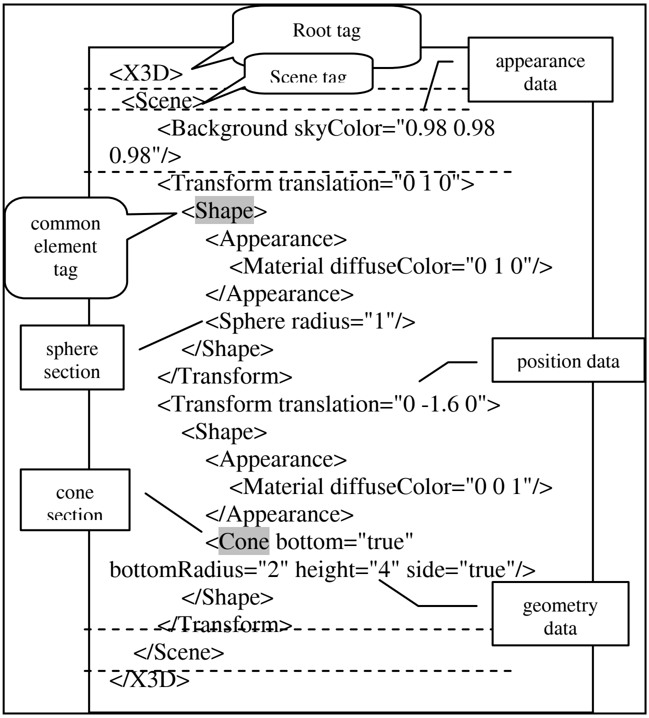
Finally, the values of these geometry elements are all recorded and belong to the particular “shape” nodes in VRML/X3D file. More values of appearance fields could also be obtained from material or color nodes. In fact, the extracted results from these unstructured data are organized as tree-like structured data after analyzed by semantic engines. The 3D feature data expressed in XML-compatible format (e.g. XMI file) will bring a new opportunity for analysis engine in feature extraction and computation. It advises that all data collection and transformation's type are transformed into the factorable standard structured document.

But it is not the only place to store analysis results for applications to use . Another classic approach is to extract annotations from a CAS and populate them into relational databases. For example, Apache Derby database is offered in Apache UIMA project as a popular open source relational database.

4 Experiments

We design the following experimental environment to implement our 3D scene analysis method, which also shows the details of how to run an analysis engine over VRML/X3D documents.

Firstly an installation of Apache UIMA release package for Windows platform is necessary. The Apache UIMA project, an implementation of the Java UIMA framework, comes with many components for different artifacts that can be created and analyzed.



```
<X3D>
  <Scene>
      <Background skyColor="0.98 0.98
0.98"/>
      <Transform translation="0 1 0">
        <Shape>
          <Appearance>
            <Material diffuseColor="0 1 0"/>
          </Appearance>
          <Sphere radius="1"/>
        </Shape>
      </Transform>
      <Transform translation="0 -1.6 0">
        <Shape>
          <Appearance>
            <Material diffuseColor="0 0 1"/>
          </Appearance>
          <Cone bottom="true"
bottomRadius="2" height="4" side="true"/>
        </Shape>
      </Transform>
    </Scene>
</X3D>
```

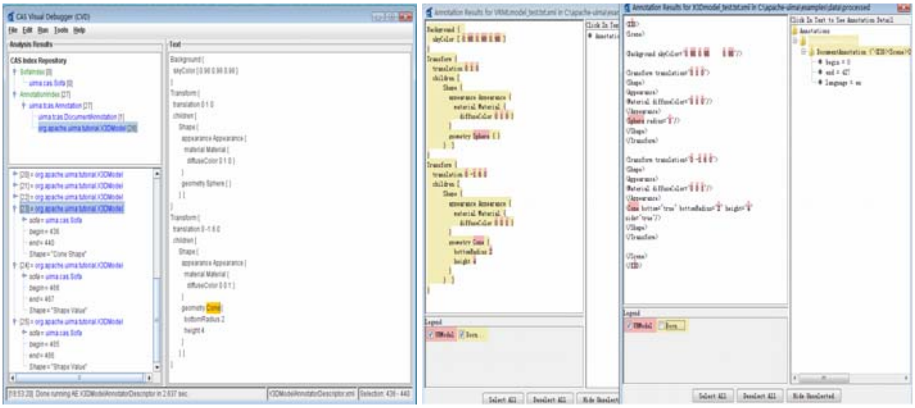
Fig. 4. Segmentation result for an example of 3D scene document

We propose UIMA tooling method to process VRML/X3D scene data, and it transforms the semi-structure graphic data into XMI format to store. A raw 3D scene document will be segmented as shown in Fig. 4. The scene is described into element levels with specific tags. When the algorithms detect these tags, the annotator will annotate the requested elements.

We design annotators to detect basic shapes following simple geometry conventions. In the scene file in Fig. 4, there are two kinds of patterns (names and values) needed to be found. And the corresponding regular expression patterns are able to be defined referring to X3D scheme. Then there are several steps to develop a simple 3D scene annotator by using UIMA framework:

Step 1. Define the CAS types for the annotator to use.
Step 2. Generate the Java classes for these types.
Step 3. Write the actual annotator Java codes.
Step 4. Create the Analysis Engine descriptor.
Step 5. Execute the annotator.

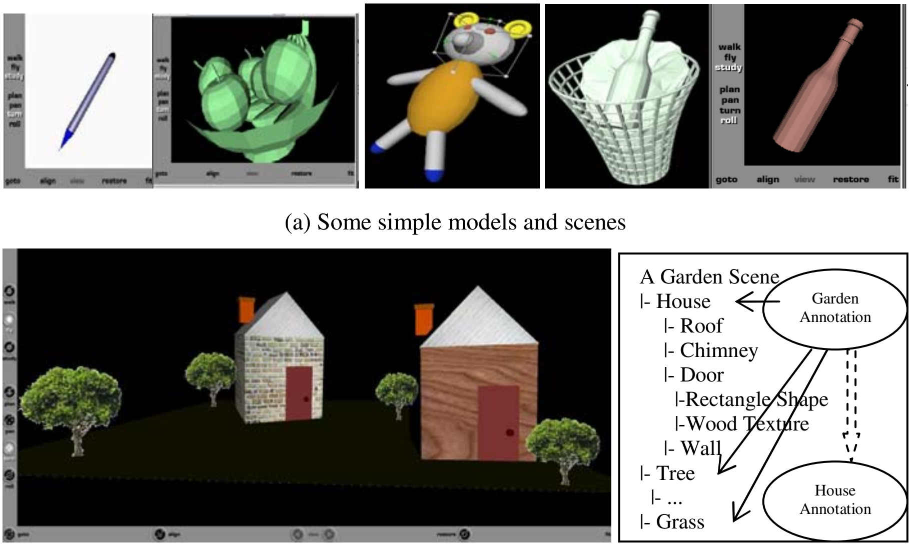
Further, to run the 3D data analysis experiments, we use the UIMA DocumentAnalyzer tool, which can run UIMA analysis components (also known as annotators) on a given set of VRML/X3D documents [2]. Fig. 5 shows the processed results.


CAS Visual Debugger (CVD)
File Edit Run Tools Help
Analysis Results
CAS Index Repository
SofaIndex [1]
uima.cas.Sofa [1]
AnnotationIndex [27]
uima.tcas.Annotation [27]
uima.tcas.DocumentAnnotation [1]
org.apache.uima.tutorial.X3DModel [26]
[20] = org.apache.uima.tutorial.X3DModel
[21] = org.apache.uima.tutorial.X3DModel
[22] = org.apache.uima.tutorial.X3DModel
[23] = org.apache.uima.tutorial.X3DModel
sofa = uima.cas.Sofa
begin = 436
end = 440
Shape = "Cone Shape"
[24] = org.apache.uima.tutorial.X3DModel
sofa = uima.cas.Sofa
begin = 466
end = 467
Shape = "Shape Value"
[25] = org.apache.uima.tutorial.X3DModel
sofa = uima.cas.Sofa
begin = 485
end = 486
Shape = "Shape Value"
Text
Done running AE X3DModelAnnotatorDescriptor in 2.637 sec
X3DModelAnnotatorDescriptor.xml
Selection: 436 - 440
Click In Text to See Annotation Detail
Annotations
DocumentAnnotation
begin = 0
end = 427
language = en
Legend
VRModel
Item
X3DModel
Select All
Deselect All
Hide Unselected

```
Background {
 skyColor [0.98 0.98 0.98]
}
Transform {
 translation 0 1 0
 children [
  Shape {
   appearance Appearance {
    material Material {
     diffuseColor 0 1 0 }
   }
   geometry Sphere { }
  } ]
}
Transform {
 translation 0 -1.6 0
 children [
  Shape {
   appearance Appearance {
    material Material {
     diffuseColor 0 0 1 }
   }
   geometry Cone {
    bottomRadius 2
    height 4
   }
  } ]
}
```

Fig. 5. Annotation results of Analysis Engine. The left one is a snapshot of VRModelAnnotator, while the right one is the generated annotations (highlighted).

Generally, the UIMA analysis results are stored as .xmi files, which describe different kinds of annotations as a (Basic Model Name, Basic Model Value) pair. The advantage is that it allows analysis results to be shared between annotators. It is also convenient for annotators to use previous results in iterated development. This means that you can label a scene step by step via iterative subdividing process. These scenes are often more complex, and constructed by many simple components, as shown in Fig. 6. They can be annotated by defined annotators, in order to segment meaningful area in color, or extract a single partition from a scene.



(a) Some simple models and scenes

(b) More complex scene

Fig. 6. 3D models and scenes. The simple model is made up of basic shapes, and complex scene is composed of various models. They can be analyzed by annotators on different scales.

Moreover, although XML is also semi-structured, the matured XML parsing tools can easily transform it into perfect structured data in relational language and then be stored. Reversely, data from database can also be easily transformed into XML format and displayed appropriately via XSLT. Therefore, the structured data are successfully generated from the semi-structured 3D scenes.

5 Conclusions

The 3D scene data is an important multimedia data type, and is highly valuable in practical applications. This paper presented an efficient approach to unify its data format and analyze it by annotators implemented using IBM's UIMA framework. In general, the analysis principle of UIMA represents a trend in multimedia data analysis that the unstructured data could be transformed into the structured one first, and then be analyzed on the basis of the generated structured indices, which would be beneficial to semantic-based multimedia analysis.

However, due to the huge data volume of 3D geometry models and the complexity of 3D scenes, there are still some difficulties for our proposed method to extract semantic features from each large 3D scene, such as memory problem, transmission problem, and exaggeration problem, and so on. How to overcome those difficulties will be the future direction of our research.

Acknowledgments. This work is supported in part by the National Natural Science Foundation of P.R. China under Grants 60875011, 60505008, 60603086, in part by the Natural Science Foundation of Jiangsu Province under Grant BK2007520, in part by the Key Program of the National Natural Science Foundation of P.R. China under Grant 60723003, and in part by IBM UIMA Innovation Award 2007. The authors would also like to thank the editors and the anonymous reviewers for their helpful comments on the earlier version of this article.

References

1. Yang, Y., Lin, H., Zhang, Y.: Content-Based 3D Model Retrieval: A Survey. IEEE Trans. on Systems, Man, and Cybernetics-Part C: Applications and Reviews 37, 1081–1098 (2007)
2. UIMA Tutorial and Developers' Guides, `http://incubator.apache.org/uima/downloads/releaseDocs/2.2.2-incubating/docs/html/index.html`
3. Web3D Consortium, `http://www.web3d.org/x3d/specifications`
4. Ferrucci, D., Lally, A.: Building An Example Application with the Unstructured Information Management Architecture. IBM Systems Journal 43, 455–475 (2004)
5. Yang, Y.B.: A UIMA Implementation of Content-based 3D Model Retrieval. Technical report, Computer Science Department, Nanjing University (2008)

Detect and Localize Faults in Alias-Free Programs Using Specification Knowledge*

Safeeullah Soomro[1] and Franz Wotawa[2]

[1] Yanbu University College
Department of Computer Sciences
Yanbu Al-Sinaiyah, Kingdom of Saudi Arabia
safee@ieee.org
[2] Technische Universit"at Graz
Institute for Software Technology (IST)
Inffeldgasse 16b/2, A-8010 Graz, Austria
wotawa@ist.tugraz.at

Abstract. Locating faults is one of the most time consuming tasks in today's fast paced economy. Testing and formal verification techniques like model-checking are usually used for detecting faults but do not attempt to locate the root-cause for the detected faulty behavior. This article makes use of an abstract dependences between program variables for detecting and locating faults in alias-free programs in cases where an abstract specification is available. The idea of using dependences for fault detection and localization is not new. But the relationship between the abstract model and the concrete evaluation of programs have not been considered so far. In particular we show that the dependence model is correct. Whenever the dependence model reveals a fault there is a test case, which also reveals a fault.

1 Introduction

Ensuring reliability of software is an important but time consuming task in today's software engineering process. Especially in the context of embedded systems or safety critical systems a high reliability has to be ensured, which requires exhaustive testing or formal verification. Although many advances have been reported in the context of testing and formal verification there is less work on localizing and repair a fault once it is revealed. In this paper, we focus on fault localization and show that techniques from verification can be used. In particular we rely on Jackson's abstract dependences approach [6] to verification. We will show that the abstract dependences cannot only be used to detect faults but also to locate them in the source code. In contrast to previous work on using abstract

* Authors are listed in alphabetical order. The research herein is partially conducted within the competence network Softnet Austria (www.soft-net.at) and funded by the Austrian Federal Ministry of Economics (bm:wa), the province of Styria, the Steirische Wirtschaftsf orderungsgesellschaft mbH. (SFG), and the city of Vienna in terms of the center for innovation and technology (ZIT).

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 379–388, 2009.

dependences for fault localization like Friedrich et al.'s work [4] we show that the computation of abstract dependences is an abstraction of the ordinary program execution. It is worth noting that Friedrich et al's work is closely related to Weiser's static slicing approach [12] as pointed out by Wotawa [15].

In testing test cases comprising inputs and expected outputs are given. The program is run on the inputs and the computed outputs are compared with the expected outputs. In case of differences the test case reveals a fault and we say that such a test case is a negative or failing test case. Otherwise, the implementation passes the test case. We call such a passing test case a positive test case. In Jackson's Aspect system [6] there are no test cases but instead a set of expected dependences between variables are specified. A dependence is a relation between variables with the following informal semantics. A variable x depends on a variable y if a value of y influences the computation of the variable x in the given program. When comparing the testing approach with the verification approach of Jackson we see that both approaches uses the program's statements to compute outputs. In testing the programming language semantics and the given input determines the computation of the output. In the Aspect system the statements determine the computation of the dependences between the variables. Hence, someone might be interested in formalizing this similarities. Moreover, there is a need to localize a fault in case of negative test cases or differences between the expected and computed dependences. In this paper we tackle these two issues, i.e., the relationship between program execution and abstract dependences and how the latter can be used for fault localization. The fault localization work is based on previous work, e.g., [10].

To motivate our work we illustrate the basic ideas of fault localization using the faulty implementation of a multiplication operation myMult given in Figure 1. The bug lies in statement 4 where the variable x is used in the right hand side expression of the assignment instead of variable y. In order to detect the fault we first have to specify the abstract dependences for the multiplication where the result should depend on both inputs. Hence, we specify that result depends on x and y which can be written as a rule: $\texttt{result} \leftarrow \texttt{x}, \texttt{y}$ or as binary relation $\{(\texttt{result}, \texttt{x}), (\texttt{result}, \texttt{y})\}$.

```
     int myMult (int x,y) {
1.     int result = 0;
2.     int i = 0;
3.     while ( i < x ) {
4.       result = result + x ; // Should be result = result + y
5.       i = i + 1; }
6.     return result;
     }
```

Fig. 1. myMult - a faulty implementation of the integer multiplication

When approximating the abstract dependences from the source code of `myMult` using the Aspect system, we finally obtain a dependence relation $\{(\texttt{result}, \texttt{x})\}$ which fails to be equivalent to the specified dependence relation. The question now is how the root-cause of this misbehavior can be found. The idea behind our approach is the following. During the computation of abstract dependences every statement has an impact to the overall dependence set. For example statement 4 says that `result` depends on `result` and `x`. When knowing the dependences of `result` before statement 4, we can extend the relation. For `myMult` the variable `result` also depends on `i` (statement 3) and a constant 0 (statement 1). The variable `i` itself depends on `i` (statement 5), `x` (statement 3) and a constant 0 (statement 2). Hence, in this case all statements fail to deliver a relation $(\texttt{result}, \texttt{y})$ and are therefore candidates for a root-cause. Let us now extend our example by introducing an additional specified dependence $\texttt{i} \leftarrow \texttt{i}, \texttt{x}$ which is said to be valid for statements 3 to 6. In this case statements 2, 3, and 5 can no longer be candidates for the root-cause because they are necessary to compute dependences for variable `i` which fulfill the specification. Hence, only 1 and 4 remain as potential root-causes.

All arguments for extracting root-causes have been done using only dependences which are computed by analyzing statements. Hence, a adapted formalization of this process which allows for reasoning about statements and their influences on the computed abstract dependences should lead to a system which extracts root-causes automatically from the source code of programs and the specified dependences. During the rest of this paper we provide a framework for this purpose which is based on model-based diagnosis [9]. Model-based diagnosis provides the means for reasoning about statements and their influences which is necessary for our purpose.

The paper is organized as follows. We start with a discussion on program abstraction using dependences. In particular we provide definitions stating correctness of such abstractions. Afterwards we discuss extensions of the abstract model that can be used directly for fault localization. For this purpose we present also a simple algorithm for localizing single faults. Finally, we conclude the paper.

2 The Dependence Model

The use of abstraction for various purposes have been suggested in qualitative reasoning [13,7,3] an area of artificial intelligence to reason about physical systems. The basic idea of qualitative reasoning is to find the right abstraction for physical systems. The idea of abstraction can be also applied to software debugging. The underlying principle of qualitative reasoning is to represent the behavior of a system not in terms of quantities but in terms of qualitative values and their relationships. The motivation came from observations of cognitive aspects of reasoning about physical systems, i.e., humans for example require no quantitative model for deriving very meaningful results about the behavior even of complex systems. A similar argument has been used in a different domain, i.e., software analysis. Mark Weiser [11] argued that programmers them-self use an abstraction of the program during debugging.

$\{y = 1\}$
1. `x = y + 1;`
$\{y = 1, x = 2\}$
2. `z = y - 1;`
$\{y = 1, x = 2, z = 0\}$
3. ...

Fig. 2. State changes during program execution

$\{\}$
1. `x = y + 1;`
$\{(x, y)\}$
2. `z = y - 1;`
$\{(x, y), (z, y)\}$
3. ...

Fig. 3. Dependences changes during program execution

Before recapitulating some related papers in debugging using abstractions of the program's behavior we briefly describe the relationship between the original execution and its abstract variant for a small example program. Consider Figure 2 where the execution of a two line program is depicted. During execution the state of the program is changed. The state in this case is represented by the values assigned to a variable. The program execution starts with a state where variable `y` has the value 1. The execution of statement 1, leads to a new state where in addition `x` has the value 2. This process continues until the end of the program is reached. The value assignments are stored in a variable environment from ENV, where ENV is the set of all functions from variables from the set of variables $VARS$ to their values from their domain DOM.

The abstract version of the execution of the same program is given in Figure 3. Instead of changing the variable environment the computed dependence relations are changed during program execution. Hence, the qualitative version of program execution only changes the semantics function. Instead of having a semantics function of the form $exec : LANG \times ENV \mapsto ENV$ (where $LANG$ is the set of possible statements in a programming language), we now have $exec_A : LANG \times DEP \mapsto DEP$ (where DEP is the set of possible dependences, i.e. $DEP = 2^{VARS \times VARS}$).

Previous work that deals with abstraction of the program's behavior include [1] and [8] where the former introduced the basic concepts and principles and the latter applied it for debugging loops in Java programs. Other work which follows the idea of using dependences include [4,14]. In contrast we use differences between dependences as starting point for debugging whereas the others make use of observations regarding the correctness or incorrectness of computed values (w.r.t. the specification).

We now formalize the abstraction and give requirements that have to be fulfilled in order to allow extracting useful information from the abstraction. We start by specifying the operational semantics of a program. For simplicity and without reducing generality we consider assignment statements, while loops, and conditional statements. The operational semantics $exec$ of such a simple language can be defined as follows where $env \in ENV$ denotes the current variable environment, and val is a function returning the value for a given environment and expression.

- **Assignments:** $exec(x \underline{=} E, env) = env'$ where $\forall y \in VARS \wedge y \neq x : env'(y) = env(y)$ and $var'(x) = val(E, env)$.

- **Conditionals**:

  $exec(\underline{if}\ E\ \underline{then}\ S1\ \underline{else}\ S2\ \underline{end\ if}, env) =$

  $$\begin{cases} exec(S1, env)\ \textbf{if}\ val(E, env) = TRUE \\ exec(S2, env)\ \textbf{otherwise} \end{cases}$$

- **While loops:**

  $exec(\underline{while}\ E\ \underline{do}\ S\ \underline{end\ while}, env) =$

  $$\begin{cases} exec(\underline{while}\ E\ \underline{do}\ S\ \underline{end\ while}, exec(S, env))\ \textbf{if}\ val(E, env) = \\ TRUEenv \qquad \textbf{otherwise} \end{cases}$$

- **Sequence of statements:**

  $exec(S\underline{;}\ , env) = exec(S, env)$
  $exec(S1\underline{;}\ S2\underline{;}\ \ldots\ Sn\underline{;}\ , env) = exec(S2\underline{;}\ \ldots\ Sn\underline{;}, exec(S1, env))$

Note that in this semantics definition the constructs related to the language $LANG$ are underlined. Given this definition we are no able to define the outcome of test cases formally. A test case is a tuple (env_I, env_o) where $env_I \in ENV$ is the input environment and $env_O \in ENV$ is the expected output environment. The output environment needs not to specify values for all variables. Given a program $\Pi \in LANG$, and a test case $tc = (env_I, env_O)$. The program Π fails the test case tc iff the expected and computed output values diverge, i.e., $\exists\, x \in VARS : exec(\Pi, env_I)(x) \neq env_O(x)$. Otherwise, the program passes the test case. We write $fail(\Pi, tc)$ if the program fails and $pass(\Pi, tc)$ if the program passes the test case. It is obvious that either the program passes or fails the test case.

The abstraction, we use is based on dependences. As already mentioned the set of dependences DEP is $2^{VARS \times VARS}$. Hence, a set of dependences is a set of pairs. Before defining the abstract semantics $exec_A : LANG \times DEP \mapsto DEP$ we define useful operations on dependences. The first operation is the composition of dependence sets.

Definition 1 (Composition). *Given two dependence relations $R_1, R_2 \in DEP$ on V and M. The composition of R_1 and R_2 is defined as follows:*

$$R_1 \bullet R_2 = \begin{array}{l} \{(x,y) | (x,z) \in R_2 \wedge (z,y) \in R_1\} \cup \\ \{(x,y) | (x,y) \in R_1 \wedge \not\exists (x,z) \in R_2\} \cup \\ \{(x,y) | (x,y) \in R_2 \wedge \not\exists (y,z) \in R_1\} \end{array}$$

Compositions are used to compute the dependences for a sequence of statements. The above definition ensures that no information is lost. The first line of the definition of composition handles the case where there is a transitive dependence. The second line states that all dependences that are not re-defined in R_2 are still valid. In the third line all dependences that are defined in R_2 are in the new dependence set provided that there is no transitivity relation. Note that this composition is not a commutative operation and that $\{\}$ is the identity element of composition.

In order to allow the direct comparison of specified dependences with the computed ones we introduce a projection operator which deletes all dependences for variables that are not of interest like internal variables.

Definition 2 (Projection). *Given a dependence relations $R \in DEP$ and a set of variables $A \subseteq M \cup V$. The projection of R on A written as $\Pi_A(R)$ is defined as follows:*

$$Proj_A(R) = \{(x,y)|(x,y) \in R \wedge x \in A\}$$

We now define the abstract semantics function $exec_A$ similar like $exec$. We assume $dep \in DEP$ and a function $vars$ returning the variables used in an expression.

- **Assignments:** $exec_A(x \underline{=} E, dep) = dep \bullet \{(x,y)|y \in vars(E)\}$
- **Conditionals:**

  $$\begin{aligned}&exec_A(\underline{if}\ E\ \underline{then}\ S1\ \underline{else}\ S2\ \underline{end\ if}, dep) = \\ &\qquad dep \bullet ((exec_A(S1,\{\}) \cup exec_A(S2,\{\})) \times vars(E))\end{aligned}$$

- **While loops:**

  $$\begin{aligned}&exec_A(\underline{while}\ E\ \underline{do}\ S\ \underline{end\ while}, dep) = \\ &\qquad dep \bullet D(E,S,\{\})\end{aligned}$$

  where the function D computes a fix-point as follows:

  $$D(E,S,dep) = \begin{cases} dep & \textbf{if } exec_A(S,dep) \times vars(E) = dep \\ D(E,S,exec_A(S,dep) \times vars(E)) & \textbf{otherwise}\end{cases}$$

- **Sequence of statements:**

  $$exec_A(S\underline{;}\ , dep) = exec_A(S, dep)$$

  $$exec_A(S1\underline{;}\ S2\underline{;}\ \ldots\ Sn\underline{;}\ , dep) = exec_A(S2\underline{;}\ \ldots\ Sn\underline{;}, exec_A(S1, dep))$$

Similar to testing we use the $exec_A$ function to determine whether a program $\Pi \in LANG$ confirms a given specification $spec \in DEP$. A program is correct with respect to $spec$ if the computed dependences are a superset of $spec$, i.e., that the program implements at least as many dependence pairs as possible. Formally, we define that a program passes the given abstract specification $spec$ iff $exec_A(\Pi,\{\})) \supseteq spec$. Otherwise the program fails implementing the specification ($fail(\Pi, spec)$). Someone might ask why the definition of comparison between the dependences computed from a program and the specified ones is based on the super-set operator and not on the equivalence operator. The reason is that the computed dependences are only an approximation. For example, for the conditional we consider both branches but maybe one is can never be executed. This is the case when the condition always evaluates to TRUE or FALSE. Hence, using $exec_A$ we might compute too many dependence relations. But even in the case if $exec_A$ do not approximate the dependences defined in a program we would use the super-set operator rather than the equivalence operator. The reason is that specification need not be complete. Hence, in most cases we are interested that a program implements the specification and maybe more.

There are two issues left. The first is regarding the fix-point computation using D. The second is the question whether $exec_A$ is correct with respect to $exec$. For D we have to show that there is always a fix-point. We do not give a formal proof but discuss arguments in favor of the existence of a fix-point. Taking into account the computation of dependences using $exec_A$ we see that the dependences are increasing. Since, there is only a finite number of variables available there must be a fix-point. For the question regarding correctness we have to define correctness first.

The abstract semantics function $exec_A$ is correct if for all programs Π which fail a specification $spec$, there exists a test case tc that reveals a failure. Formally, we state this definition as follows:

$$correct(exec_A) \leftrightarrow (\forall \Pi : \forall spec \in DEP : fail(\Pi, spec) \rightarrow \exists tc : fail(\Pi, tc))$$

$exec_A$ has to be correct. In the case $fail(\Pi, spec)$ is true, we know that there is at least one dependence (x, y) that occurs in $spec$ but not in $exec_A(\Pi, \{\})$. The influence of one variable is not correctly represented in the implementation. Hence, there must be a test case assigning a value to y and expecting a value for x after executing the program that is not equivalent to the computed one. Otherwise, the specified dependence is not needed.

In the next section, we discuss how an extended semantics function can be used for fault localization.

3 Fault Localization

In this section we show how basic ideas from model-based diagnosis and an extended version of the introduced $exec_A$ function can be used for fault localization. The basic idea behind model-based diagnosis (MBD) [9,5,2] is to use a model of a system directly to compute diagnosis candidates. The pre-requisite of MBD is the availability of a logical model of a system which comprises different components. The outcome of diagnosis is a set of components that may causes an observed unexpected behavior. In debugging for example we are interested in statements that contribute to the computation of wrong values for some variables. Hence, statements serve as components. In our case we are interested in finding statements that cause the computation of wrong dependences, i.e., dependences that are not specified. Hence, the behavior of components must be expressed in terms of dependences. Another important idea from MBD is to make assumptions of correctness or incorrectness of statements explicit. In the MBD theory a predicate AB is used to classify a component as faulty.

In order to implement MBD for debugging we first introduce a set $\Delta \in 2^{STMNTS}$ where all faulty statements are assumed to be element. Second, in case a statement is assumed to be incorrect we do not know the behavior of this statement. Instead we assume that the variable defined in this statement depend on arbitrary variables. Hence, we introduce a model-variable ξ_i for a statement i and use this to represent an arbitrary behavior. For example, if statement i defines variable x, we represent the faulty behavior using the dependence relation

(x, ξ_i) where ξ_i is a placeholder for program variables. From here on we assume that dependence relation also might comprise model-variables and we redefine the $exec_A$ relationship and call it $exec_D$, which we define using a given Δ. For all statements i that are not element of Δ $exec_D$ returns the value of $exec_A$ applied to the statement. If i is element of Δ, we define $exec_D$ as follows:

- **Assignments:** $exec_D(x \underline{=} E, dep) = dep \bullet \{(x, \xi_i)\}$
- **Conditionals:**

  $exec_A(\underline{if}\ E\ \underline{then}\ S1\ \underline{else}\ S2\ \underline{end\ if}, dep) = dep \bullet \{(x, \xi_{i(x)})|x \in def(S1) \cup def(S2)\}$
- **While loops:**

  $exec_A(\underline{while}\ E\ \underline{do}\ S\ \underline{end\ while}, dep) = dep\{(x, \xi_{i(x)}|x \in def(S)\}$

In the above definition def returns all defined variables of a sequence of statements and $i(x)$ returns a fresh index for the ξ variable. In order to find an explanation for a detected difference between the specification and the computed dependences, we have to search for a substitution for model variables ξ that fulfill the specification. Formally a substitution σ is a function which maps model variables to a set of program variables. The result of the application of the substitution σ on a dependence relation R is a dependence relation where all model variables ξ in R have been replaced by $\sigma(\xi)$. We say that a program Π fulfills the specification *spec* under assumptions Δ iff there exists a substitution σ which makes $\sigma(exec_D(\Pi, \{\})) \supseteq spec$ true. Hence, debugging is reduced to finding a subset of the set of statements that have to be assumed incorrect in order to introduce all missing dependences.

Hence, the following algorithm implements the proposed fault localization process using abstract dependences:

Algorithm diagnose()
Input: a program Π and a specification *spec*
Output: a set of diagnoses DS

```
1. Let $DS$ be the empty set.
2. Select a subset $\Delta$ from the set of statements $STMNTS$ of $Pi$.
3. If there exists a substitution $\sigma$ and $\sigma(exec_D(\Pi, \{\})) \supseteq spec$ is true under $\Delta$,
   then add $\Delta$ to $DS$.
4. If $DS$ comprises the best explanations, return $DS$ as a result.
5. If all subsets of $STMNTS$ are tested, return $DS$ as a result.
6. Otherwise, go to Step 2.
```

To conclude this section we represent our debugging approach using the `myMult` example program from (Fig. 1). We further specify that `result` depends on `x` and `y`, which is written as $\{(\texttt{result}, \texttt{x}), (\texttt{result}, \texttt{y})\}$.

The computed dependences for `myMult` are depicted in Figure4. For summarized dependences we find fix-point computation because when we add dependences line by line using iteration we find the dependences are increasing monotonically. So The

```
    int myMult (int x,y) {
1.    int result = 0; // D(1)={}
2.    int i = 0;// D(2)={}
3.    while ( i < x ) { // D(3)={(result,result),(result,x),(result,i),(i,i),(i,x)}
4.      result = result + x ; // D(4)={(result,result),(result,x)}
5.      i = i + 1; } // D(5)={(i,i)}
6.    return result; // D(6)={(result,result),(result,x),(result,i),(i,i),(i,x)}
      // Specified dependences:
      //{(result,x),(result,y),(result,result),(result,i),(i,i),(i,x)}
    }
```

Fig. 4. The computed dependences for the integer multiplication example

dependences in line 3 are the summarized dependences for the while-statement.The computation of all single diagnoses is done by checking a single statement in each step of our algorithm. We assume a statement to be incorrect and the remaining statements to be correct. Compute the dependences, ground them and compare them with the specified dependences. If a contradiction arises the assumption that the selected statement is incorrect is wrong. The computational complexity is polynomial in the number of statements provided a polynomial algorithm for computing dependences if we are only searching for single bugs.

When using our model we obtain two single fault diagnosis. Statement 1 or 4 can be the root cause of the detected differences between the dependences. For example, Statement 5 cannot be responsible for the following reasons.

Assume statement 5 to be abnormal, i.e., $Ab(5)$. From this we derive the dependence $D(5) = \{(\texttt{i}, \xi_5)\}$ which leads to the summarized dependence

$$D(3) = \{(\texttt{result}, \texttt{result}), (\texttt{result}, \texttt{x}), (\texttt{result}, \texttt{i}), (\texttt{result}, \xi_5)(\texttt{i}, \xi_5), (\texttt{i}, \texttt{x})\}$$

of statement 3. After substitution we obtain to dependence relations (`result`,`y`) and (`i`,`y`) where the former is an element of the specified dependences but the latter is not. We obtain a contradiction and conclude that our assumptions cannot be true anymore. With similar computations we can rule out statement 2 from the list of candidates. Statement 3 can also not be the a candidate because it would lead to a dependence (`i`,`y`) which is not a specified one.

4 Conclusion

In this paper we focus on debugging based on abstract dependence models. In the first part of the paper we discuss the correctness of the abstraction process. In particular we show that verification based on abstract dependence is correct with respect to the program language semantics. The correctness is explained by stating that whenever the abstract model reveals a bug there exists a test case, which also reveals the bug. The approach is not complete because it uses an approximation algorithm for computing the abstract dependence. We discuss this issue and concluded that completeness might also have undesired effects like the requirement that a specification has to be complete.

In the second part of the paper we extend the approach to fault localization. The ideas behind are to use model-based diagnosis for this purpose. We introduce the concept of model variables that handle unknown dependence relations and make incorrectness assumptions explicit. The latter is done by using a set of statements that are assumed to behave wrong. Finally, we give a non-deterministic algorithm for computing all diagnoses and conclude this part with an example.

Future research will include extensions of the model, e.g., to handle object-oriented features and to provide an empirical analysis. Another open issue is the connection with program slicing that is also based on abstract dependences.

References

1. Cousot, P., Cousot, R.: Abstract interpreation: A unified lattice model for static analysis of programs by construction of approximation of fixpoints. In: Proceedings POPL 1977, pp. 238–252. ACM, New York (1977)
2. de Kleer, J., Williams, B.C.: Diagnosing multiple faults. Artificial Intelligence 32(1), 97–130 (1987)
3. Forbus, K.D.: Qualitative process theory. Artificial Intelligence 24, 85–168 (1984)
4. Friedrich, G., Stumptner, M., Wotawa, F.: Model-based diagnosis of hardware designs. Artificial Intelligence 111(2), 3–39 (1999)
5. Greiner, R., Smith, B.A., Wilkerson, R.W.: A correction to the algorithm in Reiter's theory of diagnosis. Artificial Intelligence 41(1), 79–88 (1989)
6. Jackson, D.: Aspect: Detecting Bugs with Abstract Dependences. ACM Transactions on Software Engineering and Methodology 4(2), 109–145 (1995)
7. Kuipers, B.: Qualitative simulation. Artificial Intelligence 29, 289–388 (1986)
8. Mayer, W., Stumptner, M.: Debugging program loops using approximate modeling. In: Proceedings ECAI 2004, Valencia, Spain, pp. 843–847 (2004)
9. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence 32(1), 57–95 (1987)
10. Soomro, S.: Using abstract dependences to localize faults from procedural programs. In: Proceedings Artificial Intelligence and Applications, Innsbruck, Austria, pp. 180–185 (2007)
11. Weiser, M.: Programmers use slices when debugging. Communications of the ACM 25(7), 446–452 (1982)
12. Weiser, M.: Program slicing. IEEE Transactions on Software Engineering 10(4), 352–357 (1984)
13. Weld, D., de Kleer, J. (eds.): Readings in Qualitative Reasoning about Physical Systems. Morgan Kaufmann, San Francisco (1989)
14. Wieland, D.: Model-Based Debugging of Java Programs Using Dependencies. PhD thesis, Vienna University of Technology, Computer Science Department, Institute of Information Systems, Database and Artificial Intelligence Group (2001)
15. Wotawa, F.: On the Relationship between Model-Based Debugging and Program Slicing. Artificial Intelligence 135(1–2), 124–143 (2002)

RFID Technology and AI Techniques for People Location, Orientation and Guiding

M.D. R-Moreno[1], B. Castaño[2], M. Carbajo[1], and A. Moreno[1]

[1] Departamento de Automática. Universidad de Alcalá. Madrid, Spain
{mdolores,mcm,angel}@aut.uah.es

[2] Departamento de Matemáticas. Universidad de Alcalá. Madrid, Spain
bonifacio.castano@uah.es

Abstract. One of the main problems that we have to face when visiting public or official buildings (i.e hospitals or public administrations) is the lack of information and signs that can guide us. Thanks to the new technology advances, the electronic communication networks can be focused on an objective environment. These techniques can be used to help users to get their right location and orientation.

This is the framework we are chosen in this article. The solution proposed in this paper uses a detection and a location system based on wireless technology and Artificial Intelligence (AI) techniques to plan and inform about the paths the users can follow. The AI system is called PIPSS and integrates planning techniques and scheduling methods.

Keywords: AI Planning & Scheduling, Montecarlo method, RFID, Location & Orientation.

1 Introduction

The growth and development of information and communication technologies (ICTs) has opened a new range of applications, specially in the environment where we live. Places such as houses, offices or public institutions will be able to recognize us and to adapt to our taste, needs and preferences as soon as we come inside them. Although this can seem taken from a science fiction movie, there are already some academic and industrial initiatives that begin to show the strong current tendency to incorporate this type of techniques.

In this paper we present a solution to the problem that a user has to face when visiting large surfaces of high affluence level of people. The application is called SIGUEME (Sistema Inteligente de GUiado para Entornos Multiusuario Extensos - Intelligent Monitoring System in Big Multiuser Enviroments)[1] and it includes software and hardware elements: RFID passive technology [4] for the user detection, and an AI planning and scheduling techniques for the orientation and the guiding of the patients. Radio-frequency identification (RFID) is an automatic identification method, relying on storing and remotely retrieving data

[1] In English SIGUEME means Follow Me.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 389–398, 2009.

using devices called RFID tags or transponders. An RFID tag is an object that can be applied to or incorporated into a product, animal, or person for the purpose of identification using radio waves. RFID tags come in three general varieties: passive (require no internal battery), active (require power supply) or semi-passive (are battery-assisted) tags. In our application we have used the first type.

We have chosen a concrete example of a big surface with people going back and forward to test our system: a medical centre. The scenario is as follows. At the information desk the receptionist identifies the patient, verifies the appointment and give him an individual RFID card. Each time he goes through a RFID arch, the system detects him and calculates his position. The information to guide the patient is displayed on several screens located at geographically strategic points along the building. When looking for the place to go, the visitor can lose his way, in that case, the system will detect this situation and will correct the mistake.

The paper is structured as follows. Section 2 describes the philosophy we have followed to place the RFID sensors. In section 3 the architecture of our system is presented. Then, experimental results are shown. Finally, conclusions are outlined.

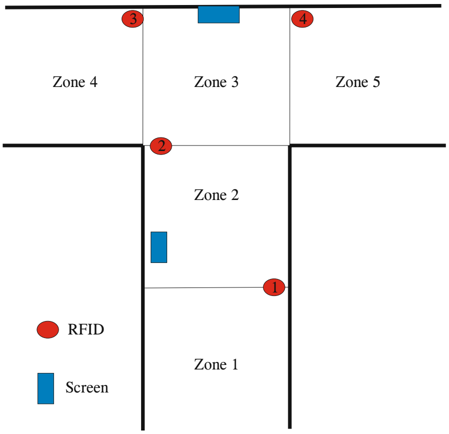
2 The Building Description

In order to follow the patients' movements inside a building, we have used a set of RFID detectors placed at strategic points. Once the RFID detectors are placed, the building gets divided into zones (rooms between the detectors). The zones are not all equal and they can have different features and functions. We have considered three different zones:

- Input Zones: are the first zones reached by the patients when they get into the hospital. They are outside any RFID detector. The information desk is placed in these zones and generally there is only one.
- Transition Zones: are intermediate zones between two or more RFID detectors. They contain the information screens.
- Destination Zones: represent the waiting rooms. Besides, when a user is on his way out, the input zones play the role of destination ones.

All of these zones are enclosed by RFID detectors in such a way that the user entrance into any zone and his subsequent exit are recorded the whole time. Our system will manage all data and it will control the situation, movements and instructions for each patient. There is not limitation on the amount of RFID cards the application can handle.

Once the building is divided into zones, we can describe it by a **graph**. In this graph, zones are the vertexes and detectors are the edges. That is, two vertices/zones are connected if there is a RFID arch between them. The graph is an undirected one and it could not be simple. We use the adjacency matrix for storing this graph in the system. But, instead of using the number of edges



	1	2	3	4	5
1		RFID1			
2	RFID1		RFID2		
3		RFID2		RFID3	RFID4
4			RFID3		
5				RFID4	

Fig. 1. a) The building layout. b) The adjacent matrix.

connecting two different vertexes, which is usually one, we use the RFID detector number between the corresponding zones.

The Figure 1a shows a sketch of this scale building layout with the detectors and message monitors. The Figure 1b shows it corresponding matrix.

3 The Architecture

In this section we describe all the elements of the SIGUEME system [2]. It is composed of the following subsystems:

- The Control Subsystem manages and controls the information and communication with the rest of the subsystems.
- The Detector Subsystem is in charge of detecting the users along the building by means of the RFID cards.
- The Reasoner Subsystem guides the user inside the building.
- The Information Subsystem allocates the information needed for the other subsystems and the users.
- The Visualization Subsystem represents and visualizes the guiding information in the corresponding screens.

3.1 The Control Subsystem

The Control Subsystem is in charge of taking the control of the other subsystems. Besides, it has to control the every day patient data and prepare a daily doctor's appointments list. It also knows the RFID available cards and the ones in use at every moment. This subsystem is operated from the information desktop. At the beginning of the day, this subsystem creates a list with all the patients with the appointments for that day. When the visitor arrives to the admission desk, the employee identifies and introduces him in the system. Simultaneously, a RFID card with a unique number is given to him. The card number is also notified to this subsystem and it identifies unambiguously its owner inside the building. The patient must keep this card the whole time inside the building.

At the same time, this subsystem finds out the patient destination and asks the Reasoner Subsystem for the track, across the hospital, for the patient. When the Reasoner Subsystem gives its answer, the Control Subsystem records it and it takes care of guiding the person towards its destination.

To deal with this task the Control Subsystem registers and distinguishes each one of the patient's RFID card. This strategy provides our application with the information about the location of all the persons inside the hospital, and even what their movements are. When a user goes through a RFID detector, the Control Subsystem calculates if he is in the right location. If a patient loses his track, the subsystem recognizes the situation and calls the Reasoner Subsystem to generate a new plan. When a person arrives to the destination point, this subsystem finishes its orientation job. If the patient leaves the hospital, he must return the RFID card and the employee will take him out of the system.

3.2 The Detector Subsystem

The aim of this subsystem is to detect the persons moving inside a building. It consists of a group of RFID detectors placed at several points in the building.

These detectors give a signal when a patient with a RFID card passes closes to them. This signal is decoded and sent to the Control Subsystem.

In this work we have developed two different detector subsystems. The first one is a hardware prototype that scales a building with one input zone, two transition zones and two destination ones. This scale model uses four RFID short range detectors and is connected to the main computer by RS232/RS485 port. This allows us to set the detectors a thousand meters away from the Control Subsystem. Reproducing the patient's movements inside the building is as easy as passing the RFID cards over the small detectors and recording the signal produced.

Although this strategy reproduces very faithfully the detection process, it is not useful at all if we want a statistically significant amount of data for testing our system. With this aim we have developed a Montecarlo simulation program that generates patient movements across the RFID detectors and works in the same way that the scale hardware model does. It has even the advantage that it can be adapted to a new building topology in a faster and easier way that the hardware prototype. Section 4.1 describes the simulation process.

3.3 The Reasoner Subsystem

This subsystem is based on AI Planning and Scheduling techniques. The Information Subsystems translates all the patient and building information into a suitable format for input to the Reasoner Subsystem. That is, the initial zone, the RFID id given to the patient, the target zone and the connections between the different parts of the hospital. So, if a person loses his way, the system is able to detect his position. At that moment, the Reasoner Subsystem is called and a new plan is generated for the lost patient.

This new plan will be translated in new screen messages managed by the Visualization Subsystem to guide the patient. The Reasoner Subsystem will produce as many times *guiding* plans as needed until each patient achieves his goal location. Finally, when a patient has accomplished his whole medical schedule, the Reasoner Subsystem will generate a new *guiding* plan to guide him right to the exit.

This subsystem is composed of PIPSS (Parallel Integrated Planning and Scheduling System) [9]. It is a system that integrates AI planning and scheduling techniques. It is based on HPP (Heuristic Progressive Planner) [10] and scheduling algorithms [3]. Its open architecture using object oriented interfaces allows the implementation and execution of different planning algorithms, scheduling methods and planning and scheduling integration schemes.

PIPSS has two kinds of planning searches: enforced hill-climbing and greedy best-first search. One type of scheduling algorithm called ISES [3] or the possibility to disable scheduling. And also, two types of planning and scheduling integration schemes: scheduling after planning or scheduling inside planning.

In addition to this, a thread search has to be instantiated with one of the three kinds of operators sets or vectors that heritages from HPP. Each one in sequence or in parallel can be launched in threats. These vectors are:

- *A vector* contains all the possible instantiated operators in the problem.
- *B vector* contains the instantiated operators generated thanks to the relaxed GraphPlan heuristic used for computing the costs in FF [8]. Each operator is relaxed by simply eliminating its delete list. The relaxed plan graph is similar to that produced by the GraphPlan, except that it does not contain any mutual exclusion relations. All the operators used in the different action levels will be collected and saved in the *B vector*.
- *C vector* is generated using an additive heuristic h_{add} for computing the heuristic cost as in HSP [1]. Considering the subgoal independence under this heuristic, *C vector* will contain all the operators that are part of the relaxed plan.

For example, if we consider the blocksworld domain with 4 possible operators (pick-up ?, putdown ?, stack ? ?, unstack ? ?); and the problem has 3 blocks (A,B,C) where blocks A and B are on the table, and block C is on top of block A, then the number of possible applicable operators in the A *vector* is 24 (this value is obtained by substituting each variable in the operators by the objects in the problem). At a first glance, we can see that there are 6 unnecessary combinations, that is, when the instantiated variables in the *unstack* and *stack* operators are equal (i.e. *(unstack A A)*, *(unstack B B)*, etc). Table 1 shows the elements of this vector.

Table 1. *A Vector* for the blocksworld domain problem

pick-up A	pick-up B	pick-up C	putdown A	putdown B	unstack C B
putdown C	stack A B	stack B A	stack A C	stack C A	unstack B C
stack B C	stack C B	unstack A B	unstack B A	unstack A C	unstack C A

Then, the B *vector* will contain 14 operators as Table 2 shows. The relaxed plan graph consists of four fact layers and three action layers.

Table 2. *B Vector* for the blocksworld domain problem

pick-up A	pick-up B	pick-up C	putdown A	putdown B	unstack C B
putdown C	stack A B	stack B A	X	stack C A	unstack B C
stack B C	stack C B	X	X	X	unstack C A

However, the C *vector* only contains **9 operators**

Table 3. *C Vector* for the blocksworld domain problem

pick-up A	pick-up B	pick-up C	putdown A	putdown B	X
putdown C	stack A B	X	X	X	X
stack B C	X	X	X	X	unstack C A

PIPSS uses PDDL [5] as the defining language for its domain and problem input files. However, PDDL lacks some characteristics for expressing temporal and multicapacity resources constraints that PIPSS can handle. Thus, it has been necessary to extend PDDL from its standard basis. The extensions have been based on the way PIPSS deals with these issues and how it describes them with its domains and problems definition language. So if a patient has different appointments in the same day, PIPSS can schedule them.

3.4 The Information Subsystem

This subsystem is in charge of keeping the whole information of the patients, the destination goals and the building information in the format understandable for each subsystem.

It uses two data sources: the building description and the patients information. It is very important that these data can be given by means of external files (although it can be introduced/modified by hand to the system), because it makes possible to change either the building or the patients information without modifying the software.

The first file basically consists of the graph building adjacent matrix, where all the zones and detectors are recorded.

The second one consists of a file with all the information about patients and their doctor's appointments. The application allows us to import this file with a predefined structured that will be saved in the database.

This subsystem also saves the following data: every signal produced by the RFID detectors, the doctor appointments for each patient, the *guiding* plans for each patient by the Reasoner Subsystem and the messages given by the Visualization Subsystem. This way we can later on analyze the system behavior and find deficiencies on the messages sent to the users, in the localization of the detectors inside the building, etc since our goal is to have a system easy to use by the user.

3.5 The Visualization Subsystem

It is in charge of sending the corresponding guiding information to the screens where the user is close to. It must be able to send the right messages to the correct places to help all the visitors simultaneously.

It is composed of a set of screens located at the building transition zones. In these devices appear the appropriate information for guiding the patients within that zone. The messages in the screen show to each user the direction he has to follow according to the *guiding* plan generated by the Reasoner Subsystem. The information that the patient visualizes is: the patient's id and the direction he has to follow. In our prototype we have chosen four different instructions: 'straight', 'turn right', 'turn left' and 'go backwards', followed by the corresponding arrow. Any other indications for the same patient will be erased from any previous display devices.

4 Experimental Results

In this section we describe the simulator developed to test the SIGUEME architecture, and the results obtained by PIPSS in the hospital domain when it is compared to other planners.

4.1 The Simulator

It is a program that simulates the patient movement through the different RFID detectors located in a concrete set of places inside a building. It is based on the Montecarlo method [11] and it only needs to know how the building zones, that is, the building graph adjacent matrix.

The purpose is to exactly provide the same detection data the hardware RFID device would supply. Doing it that way, we can easily collect a statistical significant amount of patient interactions. Software simulation allows us to avoid errors that could occur by hand simulation such as to forget any RFID card during the experimentation or to go across two different not contiguous detectors. In this first approximation we have presumed that there are no lost detections. The application also assumes the simplification that the patients, moving inside the medical centre, are able to find and understand the messages supplied by the information screens without demanding any other kind of assistance. Nevertheless, there is a certain chance that the person loses the information provided by the monitors and it had a lack of orientation that, in some cases, could get it out of the planned track. This situation can be experimented by the simulation with several levels of disorientation probabilities.

The simulator takes into account two different orientation sources for a person moving inside a building. The first one is given by the building layout and we call it "geographical orientation". This means that if a person is walking along a corridor there is a bigger probability to follow in the same direction than to go backward. In the same way, when he reaches an intersection there will be a different probability for each possible way the visitor could take according to his movement. To deal with the geographical source we have assumed a fixed probability distribution for each building zone that has two or more contiguous ones. This distribution shows the probabilities to reach each one of the contiguous ones. The second source is the set of the information screens. We can easily admit that a patient has a certain probability of misunderstand the information the system provides to him. Obviously, this situation will depend on the particular person and the monitor position. However, in our first model we have assumed that all the patients in all zones have the same probability P of getting the right direction. The two information sources are linearly combined in such a way that when P=0 we only have the geographical probability distribution and if P=100 the patient never loses his track. The visitor movements are simulated applying the Montecarlo method on that probability distribution.

The simulation program will work on demand from the Control Subsystem, until all the patients leave the medical centre.

4.2 Results

The Hospital domain is a simple path planning domain where people move among the different parts of a building. Then, the building graph adjacent matrix is used to set the problems. The action of moving from one room to another is durative, so all the people can independently move in parallel.

Four planners have been tested against PIPSS. We have also used four different PIPSS settings, so it can be said that there is a total of eight planners. They are explained as follows:

- PIPSS-A: PIPSS executing *A vector*, enforced hill-climbing, ISES and sequential search.
- PIPSS-B: PIPSS executing *B vector*, greedy best-first search, ISES and integrated search.
- PIPSS-C: PIPSS executing *C vector*, greedy best-first search, ISES and sequential search.
- PIPSS-ABC: PIPSS running three threads like the previous three configurations.
- LPG-speed [6]: non-deterministic planner LPG trying to achieve a solution as fast as posible.
- LPG-quality: non-deterministic planner LPG trying to achieve a solution with the lowest makespan (this modality cannot be launched more than once to get a better solution).

- CPT1 [12]: planning system for optimal temporal STRIPS planning with a distinguished Performance in Optimal Planning (Temporal Domains) at IPC'06.
- CRIKEY [7]: a temporal planner written in java.

We have generated 20 problems, increasing the number of persons and rooms. All executions have had a maximum available time to find a problem of ten minutes and all planners have been launched under Windows XP.

All PIPSS modalities and LPG speed found a solution to all the problems, LPG quality solved 75% of them, CPT solved 50% and, finally, CRIKEY solved 45%.

In this domain, PIPSS is the best performer (in time and makespan). All PIPSS executions find 100% of the solutions along with LPG speed, but PIPSS makespans are much better by far. LPG quality only solves 75% of the problems and its makespans are never better than those of PIPSS A or PIPSS B. The fastest searches seem to be achieved by PIPSS A and PISS ABC, both launching an EHC thread (however, CPT is a bit faster for the first ten problems, which are the only ones that it solves). The reason why PIPSS obtains the best makespans in this domain is that the EHC implementation that it uses -which comes from HPP- is very good at providing solutions with fewer steps. Since the duration of all the actions of this domain is the same (one unity of time) and since people move independently (so moving a person can be seen as a single subproblem), this means that shorter solutions will result in lower makespans. Of course, in order to do so, it is also important that the scheduler does a good job, which means that ISES performs very well when it is provided with an adequate plan, as in this case. CPT and CRIKEY only find 50% and 45% of the problems respectively but both provide good makespan, although CRIKEY is too slow.

5 Conclusions

In this paper we have presented SIGUEME, an application based on RFID and AI Planning and Scheduling techniques to solve the problem of guiding persons through large surfaces of high affluence level of people. A specific system called PIPSS has been developed for this purpose. We have tested against other state of the art planners, and PIPSS gets the best performance in makespan and time.

This system has been implemented satisfactorily in a scale prototype that has successfully proved its viability and good performance. A simulator based on the Montecarlo method has been used to test the whole architecture.

Acknowledgments

This work has been funded by the Junta de Comunidades de Castilla-La mancha project PAI07-0054-4397.

References

1. Bonet, B., Geffner, H.: Planning as Heuristic Search: New results. In: Biundo, S., Fox, M. (eds.) ECP 1999. LNCS, vol. 1809. Springer, Heidelberg (2000)
2. Castano, B., R-Moreno, M.D.: An Artificial Intelligence and RFID System for People Detection and Orientation in Big Surfaces. In: Procs. of the 6th International Conference on Computing, Communications and Control Technologies (CCT 2008), Florida, USA (2008)
3. Cesta, A., Oddi, A., Smith, S.F.: An Iterative Sampling Procedure for Resource Constrained Project Scheduling with Time Windows. In: Proceedings of the 16th Int. Joint Conference on Artificial Intelligence (IJCAI 1999) (1999)
4. Engels, D.V.: RFID: The technical Reality. In: Proceedings of Workshop on Radio Frequency Identification: Applications and Implications for Consumers, Washington, DC (2004)
5. Gerevini, A., Long, D.: Plan Constraints and Preferences in PDDL3. The Language of the Fifth International Planning Competition. Tech. Rep. Technical Report, Department of Electronics for Automation, University of Brescia, Italy (2005)
6. Gerevini, A., Saetti, A., Serina, I.: An Approach to Temporal Planning and Scheduling in Domains with Predicatable Exogenous Events. Jair 25, 187–213 (2006)
7. Halsey, K., Long, D., Fox, M.: CRIKEY - A Planner Looking at the Integration of Scheduling and Planning. In: Procs. of the Workshop on Integration Scheduling Into Planning at 13th International Conference on Automated Planning and Scheduling (ICAPS 2003), pp. 46–52 (2004)
8. Hoffmann, J., Nebel, B.: The FF Planning System: Fast Plan Generation Through Heuristic Search. Journal of Artificial Intelligence Research 14, 253–302 (2001)
9. Plaza, J., R-Moreno, M.D., Castano, B., Carbajo, M., Moreno, A.: PIPSS: Parallel Integrated Planning and Scheduling System. In: The 27th Annual Workshop of the UK Planning and Scheduling Special Interest Group (PLANSIG 2005), London, UK (2008)
10. R-Moreno, M.D., Camacho, D., Moreno, A.: HPP: A Heuristic Progressive Planner. In: The 24th Annual Workshop of the UK Planning and Scheduling Special Interest Group (PLANSIG 2005), London, UK (2005)
11. Sabelfeld, K.K.: Monte Carlo Methods in Boundary Value Problems. Springer, Heidelberg (1991)
12. Vidal, V., Geffner, H.: Branching and Pruning: An Optimal Temporal POCL Planner based on Constraint Programming. Artificial Intelligence 3 170, 298–335 (2006)

Reinforcement Learning-Based Dynamic Power Management for Energy Harvesting Wireless Sensor Network

Roy Chaoming Hsu*, Cheng-Ting Liu, and Wei-Ming Lee

Department of Computer Science and Information Engineering,
National Chiayi University, No.300, Syuefu Road,
60004, Chiayi City, Taiwan
rchsu@mail.ncyu.edu.tw

Abstract. In this study, a dynamic power management method based on reinforcement learning is proposed to improve the energy utilization for energy harvesting wireless sensor networks. Simulations of the proposed method on wireless sensor nodes powered by solar power are performed. Experimental results demonstrate that the proposed method outperforms the other power management method in achieving longer sustainable operations for energy harvesting wireless sensor network.

Keywords: Wireless Sensor Network, Dynamic Power Management, Reinforcement Learning, Energy Harvesting, Energy Neutrality.

1 Introduction

Dynamic power management (DPM) is a system-level energy utilization technique. In general cases, turning on the energy consuming device when there is no service request for the device from the system or application would induce extra energy consumption. By turning to energy saving or shutting down the idle devices before the service request comes and turning up the devices could satisfy the service demand and save energy effectively [2]. Hence, the problem of DPM could be considered as switching the device's operation status according to appropriate occasions [2]. A power management system generally has an autonomous power manager (PMR) with policy of power management as the interface or middleware between operating system (OS) and power manageable devices (PMD). The primary task of PMR is to monitor and control the operation status of PMD, such that both the criteria of energy saving and service demand could be meet. Switching among different power consumption states would induce extra energy consumption for status transition delay. Energy saving by switching among different power consumption states is effective only when the idle time is larger than status transition delay and the idle power is lower than status transition power,. Therefore, how to predict the length of idle time is a major research area in dynamic power management, which could be categorized

* Corresponding author.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 399–408, 2009.

into greedy [2], time-out [2], predictive [5], and stochastic [3, 4, 12] methods. However, recent researches on idle time prediction for DPM are mostly proposed for desktop PC with constant DC power supply, which solutions could not be completely applied to the embedded system with limited energy resources, such as cellphone, digital camera, personal navigation device (PND), etc… Besides, previous works on DPM seldom discussed power management for embedded system with renewable power source, such as solar or wind energy even thought the renewable power source is getting widely applied to embedded system. Recently, more and more research works have discussed the energy harvesting technique for embedded system, such as wireless sensor network, light electric vehicle (LEV), unmanned aerial vehicles (UAV), and satellite, etc…, where energy collectors, such as solar panel, wind-propeller generator or water-turbine, etc. [6-9, 13-15], are employed to provide the energy for embedded system. The common drawback of these renewable power sources is that the supplied power are unstable and not dependable, thus energy harvesting techniques are only be considered as secondary power supply to embedded system other than battery. Because of the unstable and uncontrollable environment, the sustainable operation of embedded system under such environment should be taken into consideration in designing the power management mechanism for energy harvesting embedded system. Kansal et al. proposed the idea of energy neutrality for embedded system, especially in the area of wireless sensor network (WSN). Kansal et al. states that the condition of energy neutrality is achieved when the energy consumption is less than or equal to the harvesting of energy [7], and that the sensor node can be continuously operated by maintaining the energy neutrality condition. Kansal further addressed the adaptive duty-cycling (ADC) method to maintain the energy neutrality condition for sensor node in WSN. In this study, a dynamic power management method based on reinforcement learning is proposed to improve the energy utilization for energy harvesting wireless sensor networks. Simulation results comparing to Kansal's adaptive duty-cycling method are demonstrated as well.

2 Architecture of Power Management for Energy Harvesting Sensor Network

Study of the dynamic power management for energy harvesting sensor node, powered by battery and renewable energy source, can be tackled by first developing an appropriate architecture such as the one shows in Figure 1. As Figure 1 shows, the architecture is constituted by two parts [10], which are system layer and hardware layer. The system layer in the upper part of the architecture is consisted of main management unit with adaptive and/or learning rules for the power management strategy and application software for sensor, communication, etc…. The lower part of the architecture is composed of the renewable energy source, energy storage unit and energy consuming units. In this architecture, the energy consumption of the sensor node, energy supply of the renewable power source and remaining energy of the energy storage device, respectively, are the three observable environment variables, denoted as e_{node}, e_{harvest}, and e_{b}, and is received by main management unit of the system layer. The main management unit decides the duty cycle of each sensor node at the beginning of sensing period, and receives the environment information of e_{node},

e_{harvest} and e_{b} at the end of sensing period. By continuously observing the environment information and adaptively controlling the operating duty cycle of the sensor node, the main management unit will gradually learns the power management policy for extending operation time of the sensor node. According to the defined architecture, dynamic power management for energy harvesting sensor node, powered by battery and renewable energy source, can be formulated as the following.

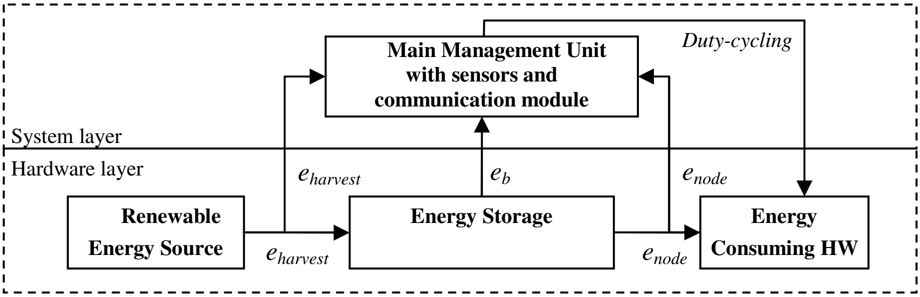


Fig. 1. System architecture for the sensor node

The energy consumption of the sensor node at the i_{th} sensing period, $e_{\mathrm{node}}(i)$, can be generally decided by the duty cycle of the i_{th} sensing period, $d(i)$, and the energy consumption of each sensor node with full duty cycle in any sensing period, e_{s}, as the following equation,

$$e_{\mathrm{node}}(i) = d(i)e_{\mathrm{s}} \tag{1}$$

In the case of energy harvesting wireless sensor network, the energy of the sensor node is supplied with renewable power source. The supplied harvesting energy in the i_{th} sensing period is denoted as $e_{\mathrm{harvest}}(i)$, and the total supply energy of renewable power source within T sensing period, denoted as $e_{\mathrm{total\ harvest}}$, is

$$e_{\mathrm{total\ harvest}} = \sum{}_{\forall i \in T}\, e_{\mathrm{harvest}}(i) \tag{2}$$

In [7], Kansal defines condition of *energy neutrality* (*EN*) as maintaining the energy equilibrium between energy consuming and energy harvesting. By maintaining the energy neutrality of a sensor node, sustainable operation of sensor node could be obtained such that the operation life of sensor is extended. The distance of energy neutrality of the sensor node, $\Delta e_{\mathrm{neutral}}(i)$, is defined as the difference between $e_{\mathrm{harvest}}(i)$ and $e_{\mathrm{node}}(i)$ at the end of the i_{th} sensing period as following

$$\Delta e_{\mathrm{neutral}}(i) = e_{\mathrm{harvest}}(i) - e_{\mathrm{node}}(i) \tag{3}$$

While, the optimal energy neutrality, e^*_{neutral}, is obtained by minimizing $\Delta e_{\mathrm{neutral}}(i)$, and is represented as,

$$e^*_{\mathrm{neutral}}\ (i) = \min\{\ \Delta e_{\mathrm{neutral}}(i),\ \forall i \in T\} \tag{4}$$

Since the harvesting energy, e_{harvest}, is practically varied time-by-time, the energy neutrality value, hence, would be bounded by the maximum and minimum energy neutrality values, represented as σ_1 and σ_2, respectively, as the following

$$\sigma_1 = \max\{ \Delta e_{\text{neutral}}(i) - \rho, \forall i \in T\} \tag{5}$$

$$\sigma_2 = \min\{ \Delta e_{\text{neutral}}(i) - \rho, \forall i \in T\} \tag{6}$$

where ρ is the average harvesting energy of $e_{\text{total harvest}}$.

3 Method Design

3.1 Reinforcement Learning

In reinforcement learning, a decision-making agent, namely the learner, takes actions in the environment the agent situated and receives reward for its actions in trying to solve a problem. After a set of trial-and-error runs, the agent should gradually learn the best policy, which is the sequence of actions that maximize the total reward." [1] The reinforcement learning model is shown in Fig. 2.

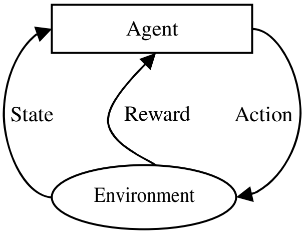


Fig. 2. Model of reinforcement learning

```
For all episodes
Initialize state s
Repeat
    Choose action a from Q (ε-greedy or soft-max)
    Take action a, observe reward r and state s'
    Update Q(s,a):
    $Q(s, a) = (1-\eta)Q(s, a)+\eta(r + \gamma\max_{\forall a`\in A} Q(s`,a`))$
    $s = s'$
Until s is terminal state
```

Fig. 3. Q-learning algorithm

For unknown environment, the agent would repeatedly count the reward during a serial of trials and the optimized policy would be obtained after a number of trials. In the beginning of the learning process, the agent is unable to make a for sure decision, the exploration strategy is, hence, required for retrieving the sufficient reward information on each state, while the exploitation is then adopted in deciding the action with higher reward and higher probability on certain state in the later part of learning process. In such a case, soft-max is utilized for the strategy of exploration-exploitation in most applications. In case of nondeterministic environment or state transition model, where the transition probability is unknown or unsure, a certain action for the next state can not be precisely decided by observing the current state. In this case, the Q-learning algorithm could be utilized in recording the accumulative reward and deciding the best policy [1].

In Q-learning, as shown in Fig. 3, the accumulative reward, Q(s, a), is function of state, s, and action, a, and the learner iteratively updates the Q-value with the updating equation shown below,

$$Q(s_t, a_t)= (1-\eta)Q(s_t, a_t) +\eta\,[r_{t+1}+ \gamma\max_{\forall \text{at}+1\in \text{A}}Q(s_{t+1},a_{t+1})] \tag{7}$$

where Q(s_t, a_t) is the accumulative reward standing at state s and take an action a at step t, and the parametersη and γ, are the learning rate and discount rate, respectively , with value between 0 and 1. r_{t+1} is a reward value which is obtained by taking action a_t and then making transition from state s_t to state s_{t+1}. Normally the Q-values are stored in a table, namely Q-table, as the reference for learner to take the next action. Fig.3 shows the Q-learning algorithm. The algorithm first initializes all environment state and Q table for each episode. In each step of episode, the action, a, is chosen according to the previous Q-table for a standing state, s, and the chosen action is executed on the system and environment. The feedback reward is immediately observed by the learner in the next step and all entries of Q table and the state are subsequently updated and the learning of each step continues until the terminal state is reached, which the next episode would start.

3.2 Dynamic Power Management for Energy Harvesting WSN Using Reinforcement Learning

In this paper, a reinforcement learning based dynamic power management for energy harvesting wireless sensor network is proposed. The agent (learner) of the reinforcement learning is employed in the main management unit of Fig. 1 to learn from the environment information of e_{node}, e_{harvest}, and e_{b}, and to adaptively decide and execute the action characterized by the desired operational duty-cycle. After executing the selected action, the agent will evaluate the performance and a reward will be calculated and granted to the agent. During the learning, the agent is encouraged to select the action with positive reward and a series of beneficial actions will be iteratively generated such that a better power management performance is gradually achieved. In the proposed method, state space of the environment, denoted as $\boldsymbol{S}$**,** is defined as a set which consisted of the state of energy neutrality, S_{D}, state of harvested energy, S_{H}, and state of the current energy storage level, S_{B}. The state of energy neutrality (S_{D}) is defined by the distance function as the following equation,

$$S_{\mathrm{D}}(i) = \Delta e_{\mathrm{neutral}}(i) \tag{8}$$

where i is the i_{th} sensing period. The state of harvested energy, denoted as S_{H} could is defined as the $e_{\mathrm{harvest}}(i)$, harvesting energy of the i_{th} sensing period, normalized to the total harvested energy in percentage. The state of energy storage (S_{B}) is defined as the percentage of the energy remained in the energy storage. Thus, the environment is constituted by different combination of states given the state space (S_{D}, S_{H}, S_{B}) $\in$ $\boldsymbol{S}$. The action space, $\boldsymbol{A}$, is defined on the controllable variable of duty-cycle, $a(i)$, that is, $\boldsymbol{A} = a(i) \in [D_{\mathrm{min}}, D_{\mathrm{max}}]$, where D_{min} and D_{max}, respectively, are the minimum and the maximum of operational duty cycle of sensor node. Hence, an action with higher value means a higher operational duty cycle is assigned to the sensor node at i_{th} sensing period; where more energy would be consumed consequently. In reinforcement learning, the reward value is utilized to evaluate the performance of decided action or policy. The distance of energy neutrality is a good candidate of reward by measuring the degree it close to zero. Thus, the immediate reward of the learning can be defined by distance of energy neutrality (r_{D}) as

$$r_{\mathrm{D}} = -|S_{\mathrm{D}}(i)| \,/\, (e_{\mathrm{maxharvest}} - e_{\mathrm{minharvest}}) \tag{9}$$

if the distance is a large value, the immediate reward would be small, whereas the distance is close to zero, a large immediate reward should be given. In the proposed system architecture, the harvesting energy is stored in the energy storage such that the remaining energy storage must be taking into consider in defining the immediate reward. By considering the effect of the remained energy in the storage, the equation of immediate reward can be modified as

$$r'_{\mathrm{D}} = -r_{\mathrm{D}}\,(1-2\,e_{\mathrm{b}}(i)\,/\,E_{\mathrm{B}}) \tag{10}$$

equation 10 indicates that if the storage remained is lower, a large positive reward would be given according to the relative positive distance of energy neutrality, and vice versa. In that criteria the positive distance of energy neutrality would induce the positive reward, while if the storage remained in higher level, the positive distance would induce the negative reward. The algorithm for the proposed reinforcement learning based dynamic power management method for energy harvesting wireless sensor networks is the following

```
1.  Initialize all Q(s,a) to zeros
2.      For every day
3.              Initialize state vector $\boldsymbol{s}_i$
4.              Repeat every timeslot $i$
5.              Choose duty-cycle action $d(i)$ from Q
6.              Adjust duty-cycle to $d(i)$
                ...wait until timeslot $i$ ends
7.              Observe $\boldsymbol{r}_{i+1}$ and $\boldsymbol{s}_{i+1}$
8.              Update Q($\boldsymbol{s}_i$ , $\boldsymbol{a}_i$):
                  $Q(s_i,a_i)=(1-\eta)Q(s_i,a_i)+\eta(r_{i+1}+\gamma\max_{a_{i+1}}Q(s_{i+1},a_{i+1}))$
9.              $\boldsymbol{s}_i = \boldsymbol{s}_{i+1}$
10.             Until $i = T$
```

The first line of the algorithm initializes all of the system parameters, such as state variables and Q-value and then proceeds to the agent's learning procedure. In this study, the solar energy is selected as renewable power source for an example such that a day can be defined as an episode, called a round hereafter. In each daily round, as starts in line 2, the algorithm goes through line 3 to line 10. The inner loop, as line 4, goes through line 5 to 10, which represents each timeslot iteration, or step. In line 5, the agent chooses an appropriate action from its power management policy according to the Q value at the current state, and then in line 6 the action is executed in the corresponding step. The agent then observes the environment's parameters, i.e., the next state and reward as in line 7. The step iterative formula could be written as line 8 according to the Q-learning, where the parameter η gradually decreases with the number of the observed state. After the Q-values are updated the next state replaces the current state. Once the available timeslot number for a daily round is reached, the next day (another round) comes up, as shown in line 10.

4 System Simulation and Experiment Result

4.1 Environment for Energy Harvesting System

In this study, a serial of experiment is conducted by simulation to validate the proposed method. The energy neutrality property is examined in the simulation to see whether it can be stably maintained by the reinforcement learning. Before the simulation can be conducted, the environment for energy harvesting system and the available configuration should be established first [8]. In this study, the solar energy is employed as the harvested energy source; where the proportion of the solar energy to the intensity of sunshine is calculated by the ideal solar energy equation [11, 16] with the altitude and azimuth of sunshine [17]. The solar panel is BSP-112 with reference to PowerUp Company [16], which can produced power up to 1W. The energy diminished probability by the atmosphere circumstances is 20%, where the harvested energy is diminished according to atmosphere circumstances by the normal distribution with mean 0.5 and the variance 0.1. Thus, the annual sunshine profile is obtained under the given parameters before the simulation is conducted. According to Kansal's energy harvesting model for deriving the certain sunshiny profile's feature [8], the average harvesting power ρ_1 is equal to 315.59mW, the deferential energies between the maxima/minima harvesting energy and the average harvesting power are calculated as 2702.58J (σ_1) and 1136.13J (σ_2), respectively. The average operational power ρ_2 is constrained by the average harvesting power, that is, ρ_2 must be less than ρ_1 at the given voltage supply of 3.6V. By the power constrain, the average ampere at the certain operational voltage is 87.66mA. According to the condition of energy neutrality [8], the capacity of the energy storage can be estimated, which is 5944mAH, from the configuration previous defined. Thus, a commercial battery with capacity of 7000mAH at 3.6V is chosen for simulation. In the follow experiments, four levels of duty-cycle rate are used for any sensing period which are 100%, 65%, 30% and 0% (as the sleep mode), and the corresponding ampere consumed are 160mA, 104mA, 48mA and 0mA respectively since the bigger duty cycle, the larger the energy is consumed.

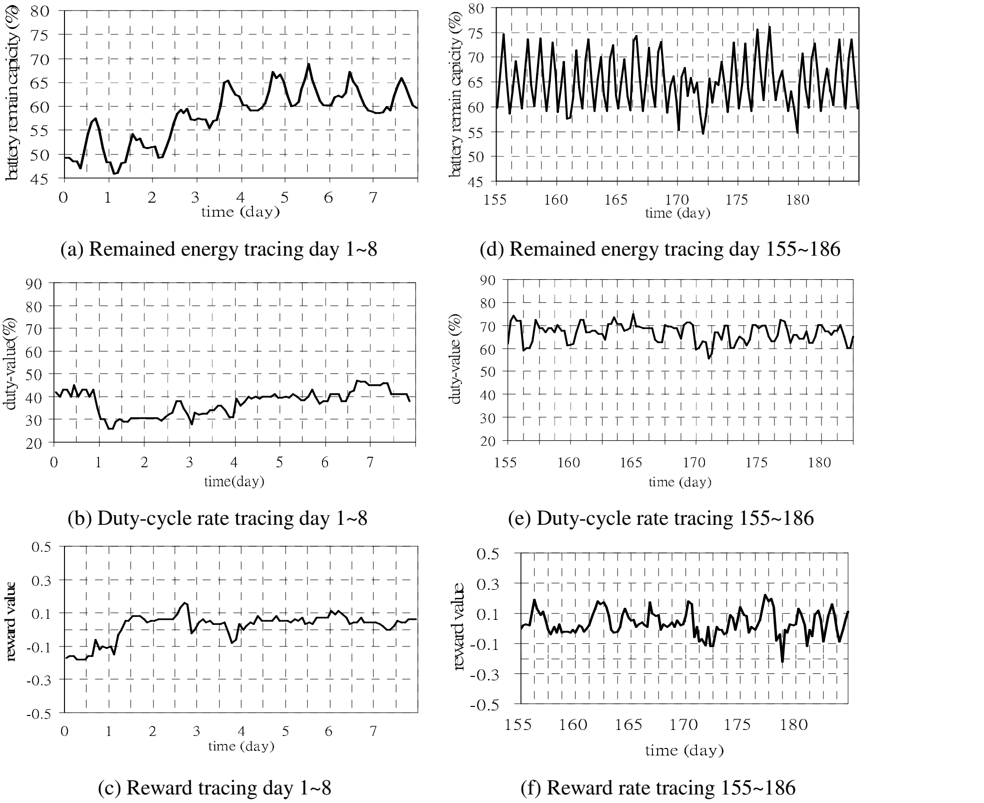
4.2 Result and Discussion

Before the simulation is conducted, the environment states, the actions of duty-cycle and the degree of energy neutrality for reward should be properly specified, such that the dimensionality of Q table in Q-learning can be defined. The state configuration for distance of energy neutrality is consisted of 3 states which are negative distance state, zero distance state and positive distance state. States of energy harvesting is consisted of 4 states, which are the ranges of 0~900J, 901~1800J, 1801~2700J and 2701~3600J. And the states of remained energy of battery is consisted of 5 states, which are in the range of 0~10%, 10~30%, 30~60%, 60~90% and 90~100%. The state configuration of actions of duty-cycle would have 4 actions which are 0%, 30%, 65% and 100%. Table 1 shows the table for reward giving, where different reward giving strategy would be used at different remained storage state. For example, if the battery remained energy is high, higher positive distance of energy neutrality will do no good for the energy storage system; hence the distance going farer form energy neutrality point positively will obtain negative reward. However, at higher storage state, the negative distance of energy neutrality should be encouraged with positive reward by increasing the duty-cycle rate to consume more energy storage.

Table 1. The rule of reward given

Storage State	distance of energy neutrality		
	<-0.5mAH	-0.5~0.5mAH	>0.5mAH
High	2	-1	-2
Middle	-2	2	-2
Low	-2	-1	2

In the simulation, 50% of remained energy in battery is used as the initial condition. The results of earlier and later simulation day are shown in Fig. 4 (a-c), and Fig. 4 (d-f), respectively.



(a) Remained energy tracing day 1~8

(d) Remained energy tracing day 155~186

(b) Duty-cycle rate tracing day 1~8

(e) Duty-cycle rate tracing 155~186

(c) Reward tracing day 1~8

(f) Reward rate tracing 155~186

Fig. 4. Simulation results based on initial battery with 50% remained energy

In Fig. 4 (a) it can be seen that the remained energy in battery gradually increases after day 1, and maintains around 60~63% after first 8 days. Fig. 4 (b) shows the daily duty-cycle profile and average daily duty-cycle rate, where the daily average is lower than 40% at the first 4 days and it is around 45% after day 4. Fig. 4 (c) indicates that the average daily reward before day 2 is negative, and after a number of learning, it is going to positive reward. After numbers of simulation days, Fig. 4 (d) shows the profile of remained energy in battery, which is around 63%. The average daily duty-cycle rate in later simulation days is shown in Fig. 4 (e), which is always higher than 55%. Because of the learning, the daily average reward in later simulation days is converged and stays around 0, as shown in Fig. 4 (f). By inspecting the simulation result above, it is found that after learning the proposed method increases the energy remaining from initial 50% to 65%, while provides above 60% of duty-cycle rate. Finally, the results of adaptive duty-cycling (ADC), proposed by Kansal [8], are compared with the results obtained using the proposed method. The simulation results of the proposed method and ADC are compared using a table, as shown in Table 2. By examining the table it can be seen that the proposed method achieves better performance in the comparing items of energy remained in the battery, which is about 2.3% increase in summer time, while maintains similar duty cycle rate. It can even be concluded that our proposed DPM method is better employed in wireless sensor network for natural animal or insect sensing in wild area during the summer time, where the activity of wild animal and insects is high.

Table 2. Experiment comparison table

Comparing items	Method		Percentage increase
	Adaptive Duty-Cycle	Proposed Method	
Ave. energy remained	54.68%	56.88%	+2.20%
Ave. energy remained (summer)	58.78%	61.12%	+2.34%
Ave. duty-cycle	54.78%	55.16%	+0.38%
Ave. duty-cycle (summer)	63.60%	63.52%	-0.08%

5 Conclusion and Future Work

In this study, a dynamic power management method based on reinforcement learning is proposed to improve the energy utilization for energy harvesting wireless sensor networks. According to the experiment result, the proposed method increases the energy remained in the storage, while maintains duty-cycle rate above 50%. By the reinforcement learning and rewarding procedure, the condition of energy neutrality is held. The findings of our study are three folds:

a. at low energy situation the agent will decrease the duty-cycle rate autonomously to reduce the energy consumption, and at the same time will store the harvesting energy as much as possible to increase the remained energy;

b. at high energy situation the agent will increase the duty-cycle rate to consume more energy so that the energy neutrality can be maintained;

c. our method maintains higher remaining energy than Kansal's adaptive duty-cycle rate method when the WSN is operating at the same average duty-cycle rate,

The proposed dynamic power management can possibly be extended to embedded system powered by other kind of renewable power source, such as wind, tide, vibration, etc….

References

1. Alpaydin, E.: Introduction to Machine Learning. MIT Press, Cambridge (2004)
2. Benini, L., Bogliolo, A., Micheli, G.D.: A Survey of Design Techniques for System-level Dynamic Power Management. IEEE Transactions on VLSI Systems 8(3), 299–316 (2000)
3. Benini, L., Bogliolo, A., Paleologo, G.A., Micheli, G.D.: Policy Optimization for Dynamic Power Management. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 18(6), 813–833 (1999)
4. Chung, E.Y., Benini, L., Bogliolo, A., Lu, Y.H., Micheli, G.D.: Dynamic Power Management for Non-stationary Service Requests. IEEE Transactions on Computers 51(11), 1345–1361 (2002)
5. Hwang, C.H., Wu, C.H.: A Predictive System Shutdown Method for Energy Saving of Event-driven Computation. In: Proc. of IEEE/ACM International Conference on Computer-Aided Design, pp. 28–32 (1997)
6. Jeong, K.S., Lee, W.Y., Kim, C.S.: Energy Management Strategies of a Fuel Cell/Battery Hybrid System Using Fuzzy Logics. Journal of Power Sources 145, 319–326 (2005)
7. Kansal, A., Hsu, J., Srivastava, M., Raghunathan, V.: Harvesting Aware Power Management for Sensor Networks. In: Proc. of ACM/IEEE Design Automation Conference, pp. 651–656 (2006)
8. Kansal, A., Hsu, J., Zahedi, S., Srivastava, M.B.: Power Management in Energy Harvesting Sensor Networks. ACM Transactions on Embedded Computing Systems 6(4), Article 32 (2007)
9. Li, D., Chou, P.H.: Maximizing Efficiency of Solar-powered Systems by Load Matching. In: Proc. of ISPLED, pp. 162–167 (2004)
10. Moser, C., Thiele, L., Brunelli, D., Benini, L.: Adaptive Power Management in Energy Harvesting Systems. In: Proc. of Design, Automation & Test in Europe Conference & Exhibition, pp. 1–6 (2007)
11. Pao, J.W.: The Evaluation of Operation Performance of a Photovoltaic System. Master thesis, Department of Electrical Engineering, National Chung Yuan University, Taiwan (2002)
12. Qui, Q., Pedram, M.: Dynamic Power Management Based on Continuous-time Markov Decision Process. In: Proc. of Design Automation Conference, pp. 555–561 (1999)
13. Raghunathan, V., Chou, P.H.: Design and Power Management of Energy Harvesting Embedded Systems. In: Proc. of ISLPED, pp. 369–374 (2006)
14. Raghunathan, V., Kansal, A., Hsu, J., Friedman, J., Srivastava, M.: Design considerations for solar energy harvesting wireless embedded systems. In: Proc. of Information Processing in Sensor Networks, pp. 457–462 (2005)
15. Zhuo, J., Chakrabarti, C., Lee, K., Chang, N.: Dynamic Power Management with Hybrid Power Sources. In: Proc. of Design Automation Conference, pp. 871–876 (2007)
16. Honsberg, C., Bowden, S.: Photovoltaics CDROM, http://www.udel.edu/Igert/pvcdrom/index.html
17. Square One WIKI - Solar Position: Calculator, http://squ1.org/wiki/Solar_Position_Calculator

Spatial Exact Match Query Based on the NA-Tree Approach in P2P Systems

Ye-In Chang, Chen-Chang Wu, and Ching-I Wang

Dept. of Computer Science and Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan, Republic of China
changyi@cse.nsysu.edu.tw

Abstract. In this paper, we propose to apply an NA-tree in the Chord system to encode spatial region data in the data key part used in the hash function to data search. That is, we combine the NA–tree with the Chord system to solve the overlapping problem which the P2PR–tree can not deal with. From our simulation results, we show that the number of visited peers in our approach is less than that in the P2PR–tree.

Keywords: Chord system, exact match query, P2P, searching, spatial data.

1 Introduction

Spatial data occurs in several important and diverse applications in P2P systems, for example, P2P virtual cities, GIS, development planning, *etc.* For the problem of answering exact queries for spatial region data in the P2P environment, an R–tree based structure probably is a good choice. Since a peer system is dynamic, the global update characteristics of data insertion/delection in an R–tree can not work well in a P2P system. Moreover, the problem of overlaps in an R–tree results in large number of the disk accesses (which will be considered as large number of messages in P2P systems). Although the P2PR–tree [1]can achieve the goal of the local update for data insertion/deletion, the overlapping phenomenon is still hard to solve.

Recently, for region data access, an NA–tree [2] has been proposed which outperforms R–tree–like data structures. It does not have the problem of overlaps which may occur in an R–tree. On the other hand, the Chord system [3] is a well–known P2P system. Since the Chord system is a hash approach, it is easy to deal with data insertion/delection with only local update. Therefore, in this paper, we propose to apply an NA-tree in the Chord system to encode spatial region data in the data key part used in the hash function to data search. Thus, we combine the NA–tree with the Chord system to solve the overlapping problem which the P2PR–tree can not deal with. From our simulation results, we show that the number of visited peers in our approach is less than that in the P2PR–tree.

The rest of the paper is organized as follows. In Section 2, we introduce the P2PR–tree. In Section 3, we present the proposed *NA–tree approach.* In Section

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 409–418, 2009.

4, we compare the performance of our approach with the P2PR–tree. Finally, we give a summary.

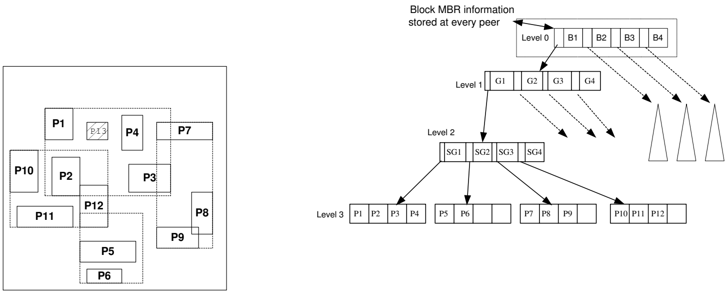
2 P2PR-Tree

Yilifu *et al.* proposed P2PR–tree strategy for object indexing in 2D–space [1] which will have only local update to the proposed index structure when data insertion/delection occurs. In their strategy, a MBR represents the region information that a peer own. In Fig. 1–(a), the MBR $P1$ represents that peer 1 owns the information of this region. As shown in Fig. 1–(b) and Fig. 2), when data $P13$ is inserted, only one path needs to be updated. Therefore, the P2PR–tree does not need global update for data insertion/deletion like the R–tree.

Although the P2PR–tree can achieve the goal of the local update for data insertion/ deletion, the overlapping phenomenon is still hard to solve. Take Fig. 1 as an example. If peer 9 wants to find the spatial region at $P12$, it needs to traverse the P2PR–tree for three branches. Because the region of $P12$ has the overlapping phenomenon, peer 9 has to search the branches which are related to the spatial region data until the spatial region data is found. Here, peer 9 needs to search the first, second, and fourth branches in the R–tree. Therefore, when the overlapping phenomenon occurs very often, it will cost much time to search the data.

3 An NA–Tree Approach

In this Section, we present how to answer spatial exact match queries in P2P systems. First, we describes the details of our structure. Next, we present our proposed algorithm for performing insertion operations. Then, we use an example to illustrate the process of the exact match.



(a) (b)

Fig. 1. An example: (a) spatial region data; (b) the P2PR-tree

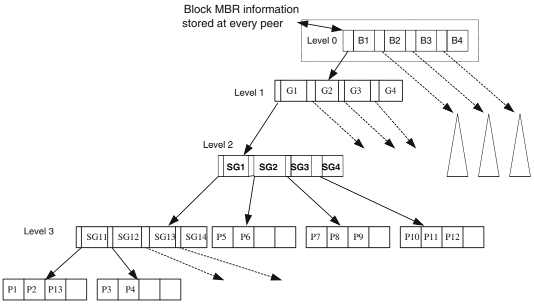


Fig. 2. An example of the P2PR–tree of a new addition data

3.1 Data Structure

In our method, we apply the NA–tree structure [2] to be as a spatial data index in P2P systems. By using the index of an NA–tree, we can assign an object to a peer in Chord. That is, we use an NA–tree approach in P2P systems. In an NA–tree, an internal node can have nine, four, or two children, and a leaf node is a terminal node. Data can be stored in an internal or a leaf node. An NA–tree is a structure based on data location and organized by the spatial numbers. First, the whole spatial region is decomposed into four regions. We let $regionI$ be the bucket numbers between 0 to $\frac{1}{4}$ $(Max\_bucket + 1) - 1$, $regionII$ be the bucket numbers between $\frac{1}{4}$ $(Max\_bucket + 1)$ to $\frac{1}{2}(Max\_bucket + 1) - 1$, $regionIII$ be the bucket numbers between $\frac{1}{2}$ $(Max\_bucket + 1)$ to $\frac{3}{4}(Max\_bucket + 1)$ $- 1$, and $regionIV$ be the bucket numbers between $\frac{3}{4}$ $(Max\_bucket + 1)$ to Max_bucket, as shown in Fig. 3–(a). Based on this decomposition, we find that when an object is lying on the space, only nine cases are possible (as shown in Fig. 3–(b)).

Nodes in an NA–tree contain index objects entries of the form $(entry\_number$, $data[1..bucket\_capacity])$, where $entry\_number$ refers to the number of objects in this node, $data[1..bucket\_bucket]$ is an array to store object data, and $bucket\_capacity$ denotes the maximum number of entries which can be stored in the node.

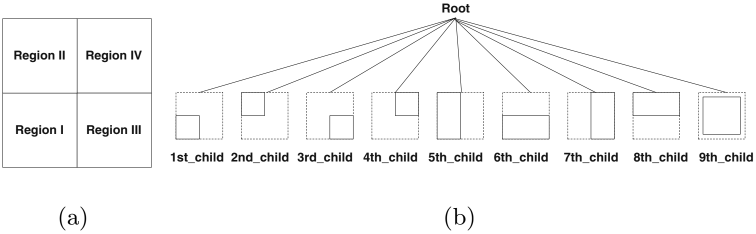


Fig. 3. The basic structure of an NA–tree: (a) four regions; (b) nine cases

```
procedure Insertion begin
    l:= The lower left coordinate of the spatial number of an object;
    u:= The upper right coordinate of the spatial number of an object;
    Find the node(nine cases) to which this object belongs at the first level
      according to the spatial number (l, u);
    Calculate the central point of the child node;
    Insert the object into a node in the NA--tree;
    if (node overflows = true)
    begin
      Split the node;
      Dispatch objects into the next level nodes in the NA--tree;
      Re-calculate the central point;
      key := Key_Method_2(p, bp);
    end
    else
      key := Key_Method_1(p, bp);
    Assign the object to the peer or peers in Chord according to the key value;
end;
```

Fig. 4. Procedure *Insertion*

3.2 The Insertion Algorithm

In this section, we describe our algorithm for inserting spatial data objects into peers in Chord. Procedure *Insertion* is shown in Fig. 4. Basically, we insert a new rectangle into a peer in Chord according the key value which is generated by the NA–tree.

Insert the Object into a node in the NA-tree. In procedure *Insertion*, the first step in inserting an object, $O(L, U)$, is to compute its spatial number, *i.e.*, the two bucket numbers of L and U. A bucket is numbered as a binary string of 0's and 1's, the so–called DZ expression. The relationship between the space decomposition process and the DZ expression is as follows [4]:

1. Symbols '0' and '1' in a DZ expression correspond to lower and upper half regions, respectively, for each binary division along the y–axis. When a space is divided on the x–axis, '0' indicates the left half, and '1' indicates right half sub–areas.
2. The leftmost bit corresponds to the first binary division, and the nth bit corresponds to the nth binary division of the area made by the $(n-1)$th division.

We use two points, $L(X_l, Y_b)$ and $U(X_r, Y_t)$, to record the region of a spatial object. Next, we calculate the corresponding bucket number of $L(X_l, Y_b)$ and $U(X_r, Y_t)$, respectively. Here, we have to convert the bucket numbers from binary to decimal form. The resulting pair of the bucket number is noted as spatial number. That is, we can use the spatial number to record an object. For convenience, we use $O(l, u)$ to denote the spatial number, where l is the bucket number

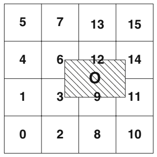
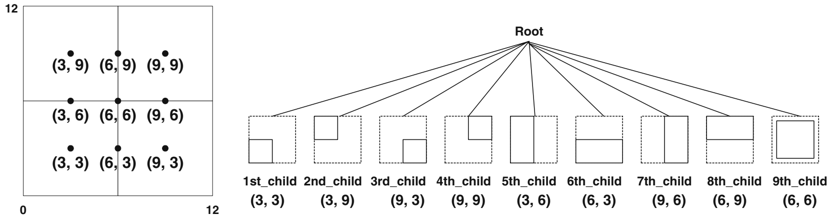


Fig. 5. An example of the bucket numbering scheme, O(l, u) = (3, 14)



(a) (b)

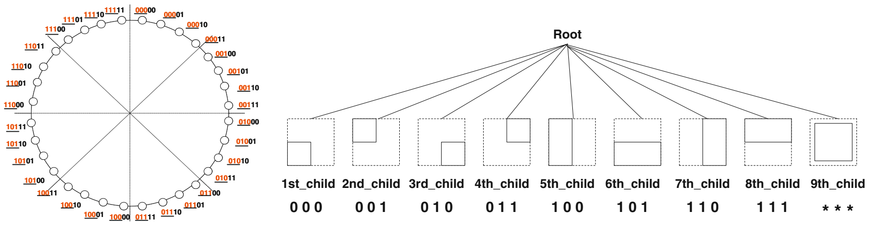
Fig. 6. An example: (a) central points of nine cases; (b) central points in the NA-tree

of $L(X_l, Y_b)$ and u is the bucket number of $U(X_r, Y_t)$. According to the spatial number (l, u), we will find the node (nine cases) to which this object belongs at the first level. For example, in Fig. 5, the spatial number of the object O is (3, 14). Since $l \in$ Region I and $u \in$ Region IV, this object belongs to 9th_child.

Then, we need to calculate the central point of the region that this object belongs to. That is, our spatial region is decomposed into four regions. Based on this composition, when an object is lying on the space, only nine cases possible. Each case (or region) has its own central point. In other words, each node in the NA–tree can be represented by its central point. Take Fig. 6–(a) as an example. The range of x–axis is from 0 to 12. The range of y–axis is from 0 to 12. The 6th_region's central point is (6, 3). Hence, the 6th_child in the NA–tree records the cental point (6, 3). The other eight children in the NA–tree can get their own central points in the same way as shown in Fig. 6–(b).

Next, this object is inserted into this node. In our NA–tree, an object is always inserted into the node at the first level. However, when a node overflows, this overflowing node is split. Then, all objects are dispatched into the next level. Thus, an object can be stored in internal or leaf nodes. After inserting the object into an NA–tree, we have to generate the key value to assign this object to an appropriate peer in Chord. We have two methods to generate the key value of an object. Basically, in both methods, first, we decide the first three bits of a data key. Next, we generate the key value of the remaining bits. Finally, we concatenate the first three and the remaining bits to get the key values of objects.

Function Key_Method_1. When an object is inserted into the first level of an NA–tree, function *Key_Method_1* that has three steps is called. First, we use three bits to represent eight cases, because the Chord ring can be split into



(a)

(b)

Fig. 7. An example: (a) eight partitions in the Chord ring; (b) the bit–expression in each region

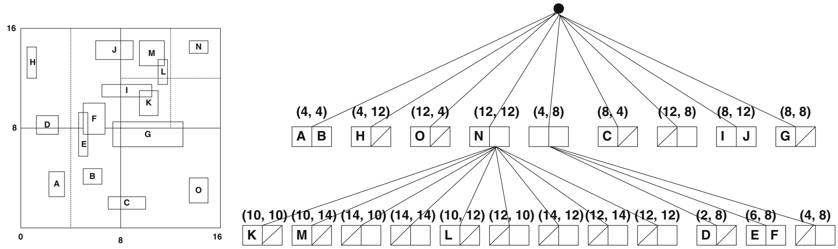
just eight partitions as shown in Fig. 7–(a). Each region in the NA–tree can be expressed by using three bits as shown in Fig. 7–(b). In particular, when the case of the 9th_child occurs, we do not use additional bit–expression to represent it. We still use the eight expression forms described above. We decide to which case the object of 9th_child should belong by the following steps: Each object of 9th_child has its own central point, and the first eight cases (or regions) have their own central point (C_x, C_y). We calculate the distance between the central point of the object and the central points of regions. Then, there will be eight results. The smallest one is our candidate. And, we can decide which case the object belong to according to the shortest distance. If there are more than one candidate, the object belongs to all of them.

Second, we generate the remaining bits by adding $(bp - 3)$ 0's. Finally, we concatenate two bit strings which are calculated by the first two steps. We know that each object must be stored in the first peer of each partition in the Chord ring, because the remaining bits are generated by adding 0's.

Function Key_Method_2. To avoid too many data to be stored in the same peer, from the second level of the NA-tree, we call function Key_Method_2 to calculate the key value of this object. There are three steps in function Key_Method_2. First, the first three bits of the key value are inherited from the node's parent. Next, we generate the remaining $(bp-3)$ bits by taking the central point of each region into consideration. Finally, we concatenate the first three and the remaining bits to get the key value of an object.

In this second step that generates the remaining $(bp-3)$ bits, we convert the decimal numbers (C_x, C_y) into binary forms, where (C_x, C_y) is the central point of each region. Next, the binary form of C_x shifts left one bit. Then, we apply the exclusive–OR operation to C_x and C_y. We can get a new binary string and choose the last $(bp-3)$ bits to be our remaining bit string.

Assign the Object to the peer. When an object is assigned to a peer in Chord, there are two buckets in each peer to store objects. One is to store objects which are owned by the peer now. The other one is to store objects



(a) (b)

Fig. 8. An example: (a) objects' distribution; (b) the NA–tree structure

which were owned by the peer before. The latter always happens when a node in the NA–tree splits and all objects are re–assigned to peers.

Now, we use one example to describe how the insertion is processed. The spatial distribution of objects is shown in Fig. 8–(a). Objects are inserted into an NA–tree in an alphabetical order. The NA–tree structure is shown in Fig. 8–(b). In our NA–tree, objects can be inserted into the internal or the leaf nodes. Because the capacity of each node is 2, the 4th and the 5th nodes need to be split. Each node has a central point's coordinate. Objects at the first level of the NA–tree use function Key_Method_1 to calculate their key values. Other objects use function Key_Method_2.

Let's explain the case of object M in details. Object M belongs to the node whose central point is (10, 14). First, we change the decimal numbers 10 and 14 to binary. We get that the binary forms of 10 and 14 are 1010 and 1110, respectively. Next, the binary form of 10 shifts left one bit resulting in 10100. Then, we will apply the exclusive–OR operation to strings 1010 (10) and 1110 (14). Finally, we can get a new binary sting, 11010, and choose the last two bits, 10, to be our remaining bit string. Finally, a key value of object M is generated by concatenating two binary strings $\underline{011}$ and 10 resulting in $\underline{011}$10. According

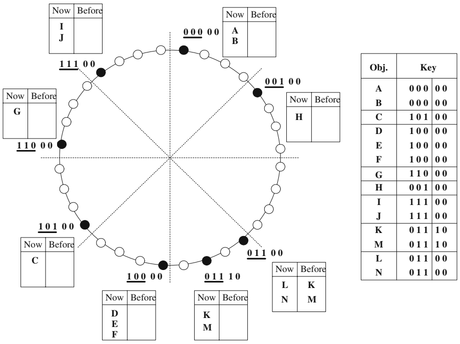


Fig. 9. An example: objects in Chord

to each object's key value, each object can be assign to an appropriate peer in Chord. Fig. 9 shows that objects are assigned to peers in Chord according to their own key values.

3.3 Answering the Spatial Exact Match Query

When we want to search a spatial object in P2P systems, first, we calculate which region this object belongs to. Then, the key value is generated by function Key_Method_1 which is described above. By using this key value, we can find a peer in the Chord ring. There are two buckets to store objects in each peer. One is to store objects it has now. The other is to record objects it had before. When we search a peer, there will be three cases:

1. The searching object is in the first bucket.
2. The searching object is in the second one.
3. The searching object is neither in the first bucket nor in the second one.

Case 1 means that we find the object and return the result, while Case 3 means that we find nothing and stop searching. In Case 2, it means that the object may be stored in some other peer in Chord. Therefore, we split the region and generate the new key value following function Key_Method_2. According to the new key value, we search the object again until we cannot find the object in two buckets of the peer.

For example, we want to find object B in Fig. 9. We get that object B belongs to the 1st_child in the NA–tree. Then, we use function Key_Method_1 to generate the key value, $\underline{000}00$. According to this key value, we search the peer, $\underline{000}00$, in Chord. We can find object B in the first bucket of this peer.

If we want to find object M in Fig. 9. We get that object M belongs to the 4th_child (*i.e.*, $\underline{011}$) in the NA–tree. Then, we use function Key_Method_1 to generate the key value, $\underline{011}00$. According to this key value, we search the peer, $\underline{011}00$, in Chord. But we can not find object M in the first bucket of this peer. However, we find object M in the second bucket of this peer. Therefore, we split the region 4 and re–calculate the key value by function Key_Method_2. Because the object belongs to the 4th region, the first three bits are $\underline{011}$. Further, the central point of the region is (10, 14). A new key value, $\underline{011}10$, is generated. According to this key value, we search the peer, $\underline{011}10$, in Chord. We can find object M in this peer's bucket one.

4 Simulation Results

In this section, we compare our approach with the P2PR–tree. Here, we define that the search cost in P2P systems is the number of visited peers [5]. Given that the data space is 1000*1000, we took measurements for six different values of the parameter P equal to 5, 6, 7, 8, 9, and 10, respectively. *That is*, there are 2^5, 2^6, 2^7, 2^8, 2^9, and 2^{10} peers in our measurements. The data objects with the average sizes 0.0025% and 0.0001% are uniformly distributed (without overlap)

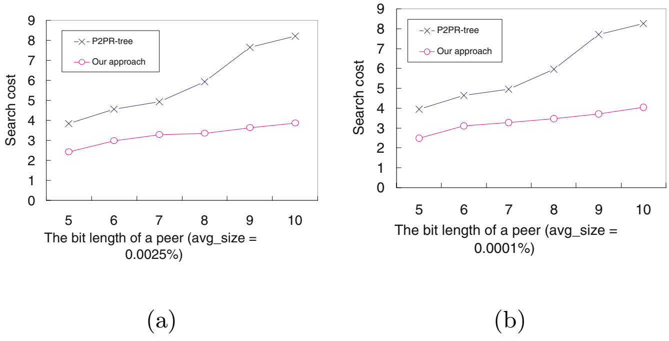


Fig. 10. A comparison of the search cost for processing a exact match query: (a) $avg\_size = 0.0025\%$; (b) $avg\_size = 0.0001\%$

on the whole data space. $Bucket\_capacity$ was assigned to be 10. For each spatial data file, we create 100 rectangles randomly to do exact match queries, and then calculate the average search cost of them.

Figure 10 shows the average search cost (in terms of the number of visited peers) of our approach and the P2PR–tree. In the P2PR–tree, a MBR is a region data as well as a peer. Hence, when the number of peers increases, the number of objects increases and causes the overlapping problem. As the number of peers increases, the search cost increases. In our approach, a MBR represents an object in the space and peers are distributed in Chord. When the number of peers increases, the objects are assigned to other peers in Chord. Therefore, as the number of peers increases, the search cost increases. From Fig. 10 we observe that our approach needs lower search cost than the P2PR–tree.

5 Conclusion

In this paper, based on the NA–tree, we have presented an approach to deal with the spatial region data in the Chord system. The Chord ring is divided into eight partitions. We use three bits to represent it. For remaining bits of a key value, we have proposed two methods to generate it by adding 0's in method 1 or taking the central point of each region into consideration in method 2. The first method is simple and applicable to the case that there are few objects in the P2P system. The second method is applicable to the case that there are too many objects in the P2P system. Then, we can get the key value by concatenating these two bit strings. According to this key value, we can assign data objects to peers in Chord. Our approach can support exact match queries in $2D$ space and reduce the overlapping problem. From our simulation results, we have shown that the number of visited peers in our approach is less than that in the P2PR–tree. Hence, our approach by using the NA–tree in the Chord system has lower search cost than the P2PR–tree.

Acknowledgments. This research was supported in part by the National Science Council of Republic of China under Grant No. NSC-97-2221-E-110-058-MY3 and National Sun Yat-Sen University. The authors also like to thank "Aim for Top University Plan" project of NSYSU and Ministry of Education, Taiwan, for partially supporting the research.

References

1. Mondal, A., Lifu, Y., Kitsuregawa, M.: P2PR–tree: An R–Tree–Based Spatial Index for Peer–to–Peer Environments. In: Proc. of Int. Workshop on Peer–to–Peer Computing and Databases, Crete, Greece, pp. 516–525 (2004)
2. Chang, Y.I., Liao, C.H., Chen, H.L.: NA-Trees: A Dynamic Index for Spatial Data. Journal of Information Science and Engineering 19, 103–139 (2003)
3. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A Scalable Peer–to–Peer Lookup Service for Internet Applications. In: Proc. of ACM SIGCOMM Conf., pp. 149–160 (2001)
4. Ohsawa, Y., Sakauchi, M.: A New Tree Type Structure with Homogeneous Nodes Suitable for a Very Large Spatial Database. In: Proc. of IEEE Int. Conf. on Data Eng., pp. 296–303 (1990)
5. Kwon, O., Moon, J.W., Li, K.J.: DisTIN – A Distributed Spatial Index for P2P Environment. In: Proc. of Data Engineering Workshop, pp. 11–17 (2006)

A Direction-Sensitive Routing Protocol for Underwater Wireless Sensor Networks

Chenn-Jung Huang[1], Yu-Wu Wang[1], Hung-Yen Shen[1], Kai-Wen Hu[1],
Po-An Hsu[1], and Tun-Yu Chang[2]

[1] Department of Computer Science and Information Engineering,
National Dong Hwa University
[2] Institute of Computer Science and Information Engineering,
National Cheng Kung University
cjhuang@mail.ndhu.edu.tw

Abstract. Underwater wireless sensor networks are expected to explore natural undersea resources and gathering of scientific data in aqueous environments. The characteristics of mobile underwater sensor networks, such as low communication bandwidth, large propagation delay, floating node mobility, and high error probability, are significantly different from terrestrial wireless sensor networks. Energy-efficient communication protocols are thus urgently demanded in mobile underwater sensor networks. In this paper, a routing protocol is developed to tackle these problems in underwater wireless sensor networks, in which fuzzy logic inference system is employed to determine the appropriate sensors to forward the packets to the destination. The proposed protocol is compared with a representative routing protocol for mobile underwater sensor networks in the literature. The experimental results verify the effectiveness and feasibility of the proposed work.

Keywords: Underwater wireless sensor networks, energy efficiency, fuzzy logic, broadcast, routing.

1 Introduction

The surface of the earth is covered about two-thirds by the ocean, and most of which are unexplored. As a promising solution to aquatic environmental monitoring and exploration, the idea of applying sensor networks in underwater environments, that is, forming underwater wireless sensor networks (UWSNs), has been attracted significant attention recently from both academia and industry [1], [2]. Wireless sensor networks have been used extensively in many land-based applications, including scientific exploration, commercial exploitation and coastline protection in the literature. Some simple techniques have been developed for underwater environment in the recent years [1].

Compared with ground-based sensor networks, mobile UWSNs have to employ acoustic communications because radio does not work well in underwater environments. Due to the unique features of large latency, low bandwidth, and high error rate, underwater acoustic channels bring many challenges to the protocol design.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 419–428, 2009.

The most limiting factor of underwater acoustic communications is the extremely low propagation speed of sound, around 1500 meters per second, subject to slight changes due to pressure, temperature and salinity variations. This is five orders of magnitude slower than the 3×10^8 m/s propagation speed of electromagnetic waves. Such high propagation delay can cause a main obstacle for interactive real-time applications and other monitoring applications, in which response time is critical.

Highly precise, real-time, and temporal-spatial continuous aquatic environment monitoring systems are extremely important for various applications, such as oceanographic data collection, pollution detection, and marine surveillance. Thus, a rapidly growing trend towards the application of sensor networks is observed in underwater environments, including monitoring, measurement, surveillance and control [3], because traditional techniques, such as remote telemetry and sequential local sensing, can not satisfy these high-demanding application requirements.

Most underwater sensor nodes, except some fixed nodes equipped on surface-level buoys, have low or medium mobility owing to water currents and other underwater activities. Underwater objects may move at speeds of 3–6 km/h in a typical underwater condition based on empirical observations [4]. Therefore, a network routing protocol that fits ground-based wireless sensor networks likely fails in the aquatic applications due to the lack of consideration of mobility for the majority of sensor nodes. Meanwhile, tiny sensors are deployed and left unattended to continuously report parameters such as temperature, pressure, humidity, light, and chemical activity. Reports transmitted by these sensors are collected by observers in the applications for ground-based wireless sensor networks. The dense deployment and unattended nature of wireless sensor networks make it quite difficult to recharge node batteries and energy efficiency is thus a major design issue in these networks.

Motivated by the above considerations, a routing protocol for UWSNs is proposed in this work to solve above-mentioned problems in underwater wireless sensor networks. The goals of the proposed protocol include satisfying the requirements of different application and improving performance metrics, such as delivery ratio, average end-to-end delay and energy consumption. The basic idea of our protocol is to constructing the routing paths for the packets based on the direction vector from source node to destination node. When a node receives a packet, it forwards or discards the packet according to its vertical distance between the node and the direction vector.

The remainder of this paper is organized as follows. Section 2 briefly reviews some related work. Section 3 presents the architecture of the proposed routing protocol. The simulation results are given in Section 4. Conclusion is made in Section 5.

2 Related Work

The sensors that are capable of forwarding and receiving acoustic wave are widely employed in underwater environment because acoustic wave is used the transmission medium in underwater environment. The deployment of the sensor networks and the techniques adopted for transmitting packets should adapt to the limitation of underwater environment, such as the size of the detection area, the number of deployed sensors, etc.

One architecture that collects data through relay sensor is the utilization of autonomous underwater vehicles (AUVs) [5]. AUVs can move without restriction of time and space. The packets are transmitted through relay sensors or are directed to the data collector at the water surface. AUVs equipped with multi-type sensors can reach any depth of ocean with higher construction and computation expense.

Another architecture of dense deployment of sensors that collects and transmits data was proposed in [6], [7]. These sensors can be settled at any specific depth in underwater environment in order to transmit packets from the bottom of water through relay sensors to the data collector on the water surface. The topology of a sensor network is usually dynamic as sensors drift passively with water currents. The sink on the water surface is equipped with both radio-frequency and acoustic transmitters to further forward the packets to the control center ashore through radio-frequency technique.

A depth-based routing (DBR) protocol was proposed in [8]. DBR does not require full-dimensional l transmission of packets. It was reported that energy consumption for DBR linear grows with number of nodes, and long transmission time and high energyocation information of sensor nodes. Instead, it needs only local depth information, which can be easily obtained with an inexpensive sensor that can be equipped in every underwater sensor node. A key advantage of DBR protocol is that it can handle network dynamics efficiently without the assistance of a localization service. It can take advantage of the multi-sink underwater sensor network architecture without introducing extra cost.

In DBR protocol, sensor nodes with acoustic modems are distributed in the interested 3-D area to collect or relay data to the sinks. When a sensor receives a packet from its neighbor sensor, it will wait a period of time to broadcast packets to each sensor within its transmission range in full dimension. The length of waiting time is changed by its vertical depth measured from water surface, and the time decreases with the increase of the vertical depth of the sensor. If the sensor receives a packet from a shallower sensor node, it discards the packet because it is aware of the existence of a fitter sensor. It is obvious the need of waiting before forwarding packets at each sensor node results in extra delay during the consumption is observed under dense deployment. These issues will be addressed in our proposed routing protocol as discussed below.

3 Architecture of Direction-Sensitive Routing Protocol

The architecture of underwater wireless sensor network adopted in this work is shown in Fig. 1. The source node is located at bottom of the water. The data collected by the source is forwarded through other sensors to the sink at the water surface. Equipped with both radio-frequency and acoustic modems, the sink receives acoustic signals forwarded from the sensors and transmits the packets to the control center ashore through radio-frequency signal.

Fig. 2 depicts the architecture of the proposed direction-sensitive routing protocol (DSR), which consists of two modules, including fuzzy logic inference system and broadcast tree growth-restricting mechanism. The fuzzy logic inference system utilizes the three parameters and selects at most two candidate sensor nodes to

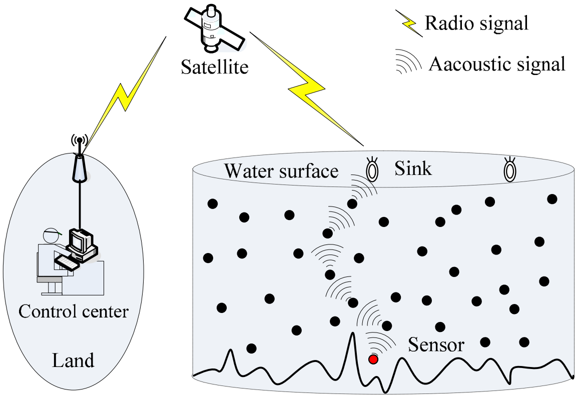


Fig. 1. Deployment of underwater wireless sensor network

forward the packets based on a fuzzy rule base. The three parameters of the fuzzy logic inference system include the distance and the angle between two neighboring sensor nodes, and the remaining energy left in the sensor node. The broadcast tree growth-restricting mechanism prevents fast spreading of packet broadcast over the UWSNs and causes unnecessary power consumption of the sensor nodes. The depth of the broadcast tree keeps under surveillance. The sensor nodes located deeper than a specific tree level stop forwarding packets a most appropriate sensor node. The specific tree level used to stop broadcasting is determined by the experiments.

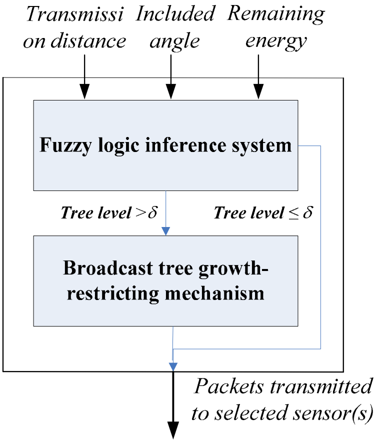

Transmissi on distance
Included angle
Remaining energy
Fuzzy logic inference system
Tree level > δ
Tree level ≤ δ
Broadcast tree growth-restricting mechanism
Packets transmitted to selected sensor(s)

Fig. 2. Architecture of the proposed direction-sensitive routing protocol

3.1 Fuzzy Logic Inference System

Fig. 3 shows the fuzzy logic system of three input variables, the function of each module is described as follows:

- **Fuzzifier:** The fuzzifier performs the fuzzification function that converts three inputs into suitable linguistic values which are needed in the inference engine.
- **Fuzzy Rule Base:** The fuzzy rule base is composed of a set of linguistic control rules and the attendant control goals.
- **Fuzzy Inference Engine:** The inference engine simulates human decision-making based on the fuzzy control rules and the related input linguistic parameters.
- **Defuzzifier:** The defuzzifier acquires the aggregated linguistic values from the inferred fuzzy control action and generates a non-fuzzy control output, which represents the candidate sensor(s).

The input-output mapping for the fuzzy logic inference system established at the *i*th sensor can be expressed by,

$$P = \mathrm{fl}_i\left(D, A, R\right) \tag{1}$$

where the inputs *D* and *A* denote the transmission distance and the angle between the *i*th sensor and its neighboring node, and R is the remaining energy left at the *i*th sensor node.

Three linguistic term sets, “Short”, “Medium” and “Long” are used for transmission distance *D*, and “Small”, “Medium” and “Big” are used for the angle *A*, and “Low”, “Medium” and “High” are used for the remaining energy *R*. The output parameter of the inference engine refers to the appropriateness of the sensor as an intermediate node for forwarding the packets to the destination. The range of the output falls between 0 and 1.

Fig. 4 shows the defuzzification process in the fuzzy logic inference system employed in this work. The non-fuzzy output of the defuzzifier can then be expressed

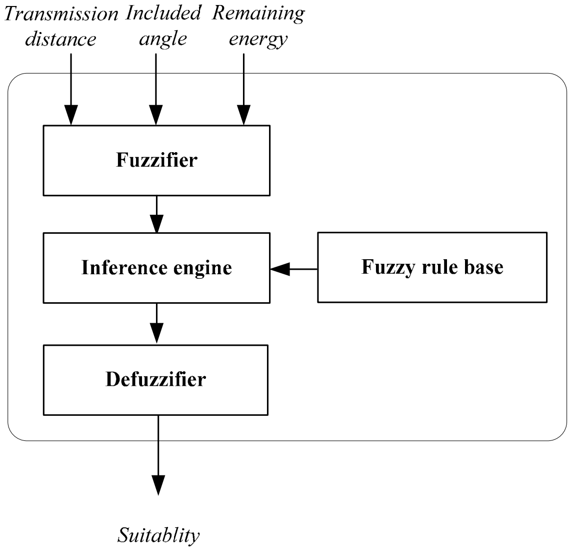


Fig. 3. Fuzzy logic system of three input variables

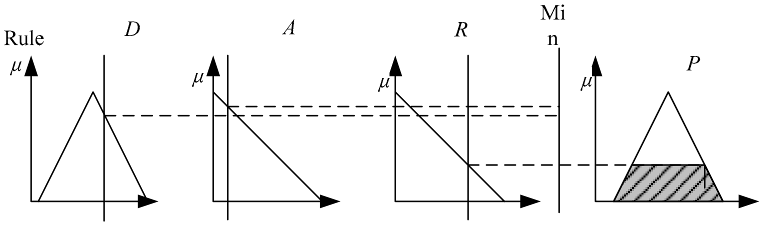


Fig. 4. Reasoning procedure for Tsukamoto defuzzification method

as the weighted average of each rule's output after the Tsukamoto defuzzification method is applied:

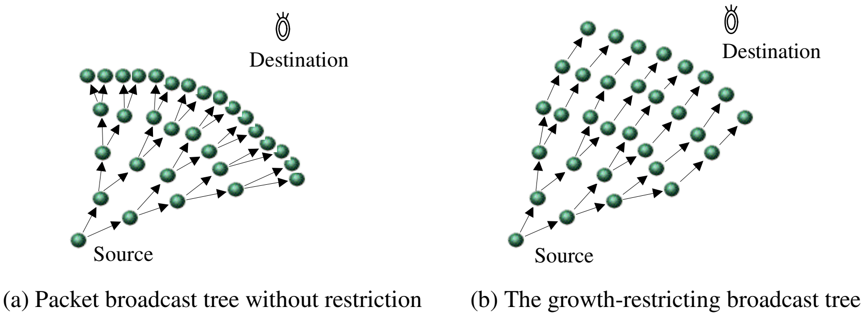
$$P = \frac{\sum_{i=1}^{18} A_i \cdot w_i}{\sum_{i=1}^{18} w_i} \tag{2}$$

where A_i denotes the output of the ith rule induced by the firing strength w_i. Notably, w_i represents the degree to which the antecedent part of the ith fuzzy rule constructed by the connective "AND" is satisfied.

3.2 Broadcast Tree Growth-Restricting Mechanism

It is observed that the fuzzy logic inference system adopted in the proposed routing protocol has some advantages, including high delivery ratio and low end-to-end delay, but with significantly high energy consumption. We believe high energy consumption in packet transmission is caused by unrestricted duplication of packets following down the broadcast tree. Instead of forwarding the packets to the best two neighbor sensor nodes all the time, we restrict the horizontal spanning of packets when the broadcast tree grows larger than a specific tree level in order to lower energy consumption.

Fig. 5 (a) illustrates the broadcast tree that grows without any restrictions, whereas Fig. 5 (b) shows the trimmed packet broadcast tree that restricts the horizontal spanning when the tree level is equal to or larger than three. It can be seen that the packets are forwarded to a signal node instead of two neighbor sensor nodes when the



(a) Packet broadcast tree without restriction (b) The growth-restricting broadcast tree

Fig. 5. The effect of broadcast tree growth-restricting mechanism

tree level is equal to or larger than three and the power consumption of the sensors nodes are accordingly reduced. The tree-level that is used to stop horizontal spanning of packet tree is determined by the experiments, which will be described more detail in Section 4.

4 Simulation Results

A series of simulations were conducted to evaluate the performance of the proposed work by using a network simulator written by C++. Each node collects the data and sends the resultant data to the sink node. The compared schemes during the simulations include the proposed direction-sensitive routing protocol (DSR) and the representative routing protocol in underwater environment in the literature, depth-based routing (DBR) protocol [8].

The underwater simulation environment setup was similar to that established in [8]. The simulations were performed for 1000 iterations. The topology deployed in each run of the simulations was randomly generated. 1000 nodes were randomly deployed in a 3-D area with dimensions$500\times 500\times 500(m^3)$. The communication range for both the sensors nodes and the sink node was 100 meters. The bandwidth of the data channel was set to 4kbps. Each data message packet was 50 bytes long and the surface sink was set at the center of water surface. The sinks are assumed to be stationary once deployed, and the sensor nodes follow the random-walk mobility pattern. Each sensor node randomly selects a direction and moves to the new position with a random speed between the minimal speed and maximal speed, which are set to 1 m/s and 5 m/s respectively unless specified otherwise. Based on the energy consumption model presented in [9], the least transmission power required at the transmitter site to achieve a power level, P_o , at the receiver can be expressed by,

$$E_{tx}(d) = P_0 \times d^2 \times 10^{\frac{\alpha(f)}{10}} \tag{3}$$

where f denotes the frequency, $\alpha(f)$ is a medium absorption coefficient depending on the frequency range of interest under given water temperature and water salinity, and d denotes the distance between the transmitter and the receiver.

As mentioned in Section 3, the way of packet transmission is broadcasting to the best two nodes within the transmission range of the sensor node that needs to forward the packets. Since the construction of packet broadcast tree may introduce costly energy consumptions, we thus investigated whether it is helpful to stop broadcasting after forwarding the packets down through several levels of the broadcast tree without seriously deteriorating the performance metrics including packet delivery ratio and end-to-end delay.

Fig. 6 to Fig. 7 show the changes of the packet delivery ratio, energy consumption and average end-to-end delay with different tree levels that the tree stops growing. The tree level, ***TL***, is varied from zero to five in these figures. Notably, ***TL*** = 0 represents the case that the tree can grow without any restriction. It is observed that packet delivery ratio can be improved with a larger tree-level value, whereas results in significantly increased energy consumption, and vice versa. In addition, it can be seen from the three figures that packet delivery ratio is substantially decreased when the tree level is down from three to two, the energy consumption is extensively increased when the tree level grows from three to five, and the average end-to-end delay is almost irrelevant to the change of the tree level. Based on the above observations, it is concluded that the overall system performance is the best when the tree level is three. We thus set the tree level that the broadcast tree stops growing to three in later simulations.

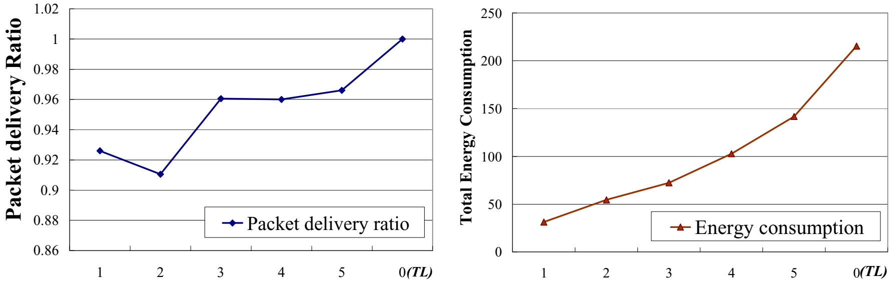


Fig. 6. Packet delivery ratio and energy consumption with different tree levels that the tree stops growing

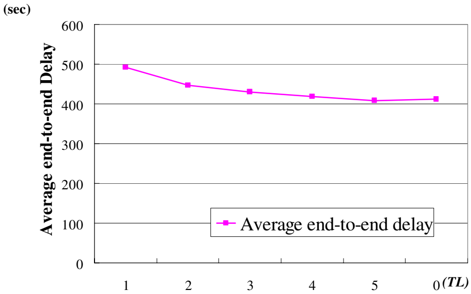


Fig. 7. Average end-to-end delay with different tree levels that the tree stops growing

Fig. 8 and Fig. 9 show the comparisons of packet delivery ratio and average end-to-end delay for DSR and DBR schemes, respectively. It can be seen that the performance of DSR and DBR are similar in packet delivery ratio, while the proposed DSR significantly outperforms DSR in terms of average end-to-end delay. We believe that the holding time required in DBR approach in order to wait a period of time to forward packets after each sensor node receives packets from its neighbor node results in higher delay time.

Fig. 10 illustrates the comparison of energy consumption for both routing protocol. It can be seen that the energy consumption of DSR is slightly increased when the number of sensor nodes gets larger, owing to the effectiveness of the mechanism restricting the growing the packet broadcast tree in the proposed protocol. This also verifies the high scalability of the DSR in terms of energy consumption.

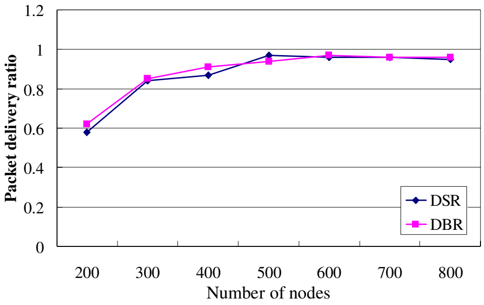


Fig. 8. Comparison of packet delivery ratio for DSR and DBR

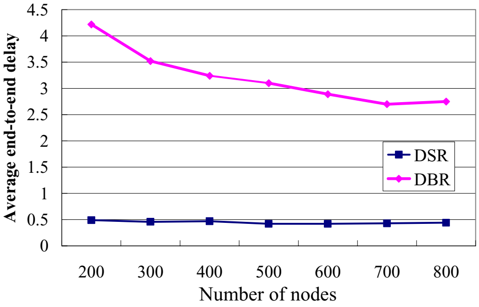


Fig. 9. Comparison of average end-to-end delay for DSR and DBR

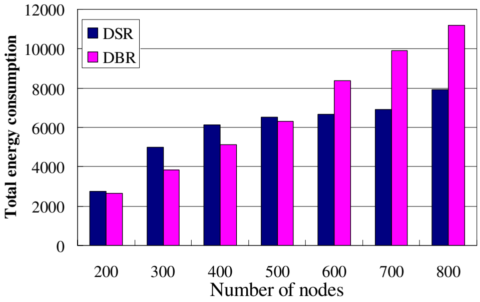


Fig. 10. Comparison of energy consumption for DSR and DBR

5 Conclusion

A direction-sensitive routing protocol for UWSNs is proposed in this work. The design of the routing protocol focus on enhancing performance metrics in volatile underwater environments, especially the need for ease of deployment and the severe energy constraints of the nodes are thoroughly considered. A fuzzy logic inference system is utilized to select the suitable sensor(s) for forwarding packets, and a simple algorithm is developed to prevent the growing of the packet broadcast tree so as to effectively reduce the energy consumption of the sensor nodes. The simulation results show that the proposed routing protocol can achieve excellent performance in terms of the metrics, the packet delivery ratio, energy consumption, and average end-to-end delay.

Acknowledgement

The authors would like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract No. NSC 96-2628-E-259-022-MY3 and NSC 97-2218-E-259-005.

References

1. Sozer, E.M., Stojanovic, M., Proakis, J.G.: Underwater Acoustic Networks. IEEE Journal of Oceanic Engineering 25, 72–83 (2000)
2. Akyildiz, I.F., Pompili, D., Melodia, T.: Underwater Acoustic Sensor Networks: Research Challenges. Ad Hoc Networks Journal (Elsevier) 3, 257–281 (2005)
3. Akyildiz, I.F., Pompili, D., Melodia, T.: Challenges for Efficient Communication in Underwater Acoustic Sensor Networks. ACM SIGBED Review 1(1) (2004)
4. Partan, J., Kurose, J., Levine, B.N.: A Survey of Practical Issues in Underwater Networks. In: Proceedings of ACM International Workshop on UnderWater Networks, pp. 17–24 (2006)
5. Akyildiz, I.F., Pompili, D., Melodia, T.: State-of-the-Art in Protocol Research for Underwater Acoustic Sensor Networks. In: Proceedings of ACM International Workshop on UnderWater Networks (WUWNet), Los Angeles, CA (2006)
6. Heidemann, J., Li, Y., Syed, A., Wills, J., Ye, W.: Research Challenges and Applications for Underwater Sensor Networking. In: Proceedings of the IEEE Wireless Communications and Networking Conference (2006)
7. Cui, J.H., Kong, J., Gerla, M., Zhou, S.: The Challenges of Building Scalable Mobile Underwater Wireless Sensor Networks for Aquatic Applications. IEEE Network (2006)
8. Yan, H., Shi, Z.J., Cui, J.H.: DBR: Depth-Based Routing for Underwater Sensor Networks. In: Proceedings of IFIP Networking 2008, pp. 72–86 (2008)
9. Park, M.K., Rodoplu, V.: UWAN-MAC: An Energy-Efficient MAC Protocol for Underwater Acoustic Wireless Sensor Networks. IEEE Journal of Oceanic Engineering 32(3), 710–720 (2007)

A Mobility and Bandwidth Prediction Controller Using an Adaptive Neuro-Fuzzy Inference System for Mobile Ad Hoc and Sensor Networks

Jerzy Martyna

Institute of Computer Science, Jagiellonian University, ul. Lojasiewicza 6,
30-348 Cracow, Poland
martyna@softlab.ii.uj.edu.pl

Abstract. In this paper, we propose a new fuzzy neural controller for a mobility and bandwidth prediction in mobile ad hoc and sensor networks. The bandwidth prediction is based on an Adaptive Neuro-Fuzzy Inference System (ANFIS) that we realized. The built controller is adaptive in the sense that it can learn and predict future location and bandwidth requirements for mobile nodes. As a result, a significant reduction of computational time was obtained. The performance of our controller was evaluated using the mobility data. The performance measure of our controller shows that the prediction of movement and the required bandwidth have a high accuracy ratio regardless of the speed of mobile nodes.

Keywords: Intelligent systems, ANFIS system, wireless communication.

1 Introduction

Mobile wireless networks have become increasingly popular in the network industry. These networks can provide mobile users or autonomic nodes with ubiquitous communication capability and information access regardless of locations. Mobile ad hoc networks are a kind of mobile wireless networks [1]. The nodes are free to move randomly and organize themselves arbitrarily. Thus, the networks topology may change rapidly and unpredictably. Mobile wireless sensor networks [10] are a particular type of mobile ad hoc networks, in which the nodes are "smart sensors", that is, small devices equipped with sensing functionalities (pressure, acoustic, etc.), a small processor, and a short-range wireless transceiver. These networks are used for data harvesting from the environment, and building a global view of the monitored region, which is accessible to the external user through the gateway node.

One of the key problems for network designers within mobile ad hoc and sensor networks is the node mobility estimation and prediction. Since these networks are connection-oriented technologies, they contain both a connection establishment phase to data exchange and a connection release phase after data exchange. Additionally, as nodes move, some of connections have to be torn down and reestablished with a frequency that corresponds to the speed of the mobile node. Moreover, the Quality-of-Service (QoS) parameters must be maintained regardless of the node's mobility.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 429–438, 2009.

To the previous works in the area of mobility prediction belongs Tabbane's [11] proposal. In this paper the mobile's location may be determined by the quasi-deterministic mobility behaviour represented as a set of movement patterns. In the paper by Hong and Rappaport [2] a mobility model based on the assumption that users have a mean movement velocity and direction in each cell was proposed. A *mobile motion prediction* (MMP) algorithm was proposed by Liu and Marquire [6]. The main drawback of the MMP algorithm is its high sensitivity to so called "random movements". In the paper by Liu [7] the next prediction using the only current location is presented. The past location information for the motion prediction is rejected. It is the major weakness of this method.

Among various combination of methodologies in "soft" computing, the one that has highest visibility at this juncture is that of fuzzy logic and neurocomputing, leading to so-called neuro-fuzzy systems. Within fuzzy logic, such systems play a particularly important role in the induction of rules from observations. One effective method developed by Jang for this purpose is called ANFIS (Adaptive Neuro-Fuzzy Inference System) [3], [4].

In this paper, we introduce a new method for mobility and bandwith prediction in mobile wireless ad hoc and sensor networks. In our approach we used a neuro-fuzzy inference system which by use history of connections can predict the future location of the direction of movement. Additionally, the proposed method allows us to compute the required bandwidth and basic QoS parameters.

The rest of the paper is organized as follows. In section 2, we describe the neuro-fuzzy inference model. Section 3 provides the proposed movement model using a neuro-fuzzy inference model for mobile ad hoc and wireless sensor networks. Section 4 gives the simulation results. Finally, we conclude with some discussion in section 5.

2 Mobility Model and Bandwidth Alocation

In this section, we describe the proposed movement model and the basic principles for the bandwidth allocation scheme.

Our movement model is built as a two-level hierarchy (see Fig. 1). The first layer (primary) represents all the nodes which are responsible for the data transmission to the sink. It consists only of clusterhead nodes and nodes belonging to the so called *backbone network*. The secondary layer is composed of ordinary nodes which collect the information and send it to the clusterheads. In our model, some nodes move or the sink is in move in relation to the static network. We assumed that all the data delivered by mobile nodes are sent to clusterheads, while the ordinary static nodes transmit their packets within their cluster.

For the mobile nodes handoffs are admitted between clusterheads. If the transmission channel is not available for the mobile node, the handoff is broken. On the other hand, if the signal received by a mobile node from the clusterhead is getting smaller and it is impossible to open a new connection, the existing link is closed.

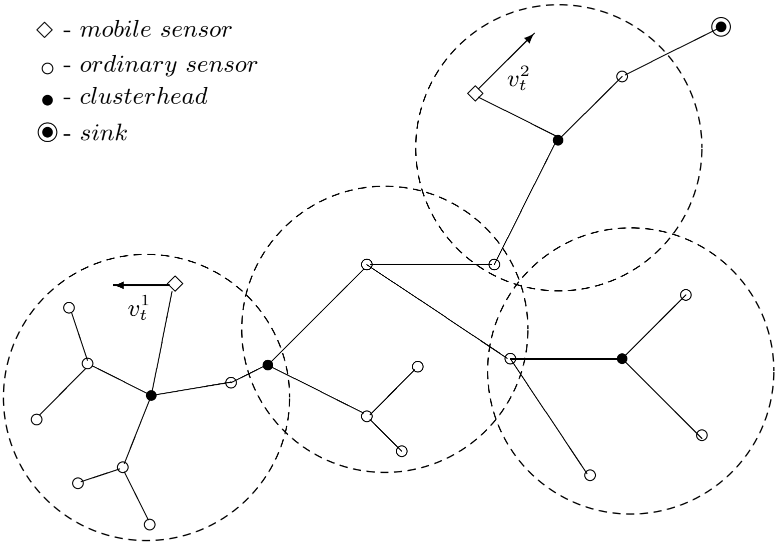


Fig. 1. An example of mobile sensor network

The submission process of a new connection is caused by one of the three events: a) the mobile node will send its data, b) the static node desires their data, c) the mobile node must execute the handoff. The liquidation of a connection is possible only in four situations: a) the termination of data transmission from the mobile node, b) the termination of data transmission from the static node, c) the termination of handoff for the mobile node, d) the loss of connection by the mobile node in the case of lack of radio range.

Our bandwidth allocation mechanism admits some classes of transmission. Depending on the transmission class and the actual channel link capacity the bandwidth C_j $(j = 1, 2, \ldots, J)$ is admitted, where J is the total number of connections. For all of connections J the following condition must be satisfied

$$\sum_{j=1}^{J} C_j \leq C_{link} \tag{1}$$

where C_{link} is the bandwidth of the link. A pool of the bandwidth C_f is defined as

$$C_f = C_{link} - \sum_{j=1}^{J} C_j \tag{2}$$

In the case of lack of free bandwidth each clusterhead borrows the bandwidth from the neighbouring clusterhead. Additionally, thanks to the movement prediction the needed bandwidth in all neighbouring clusterheads is reserved in the direction of the movement of mobile nodes.

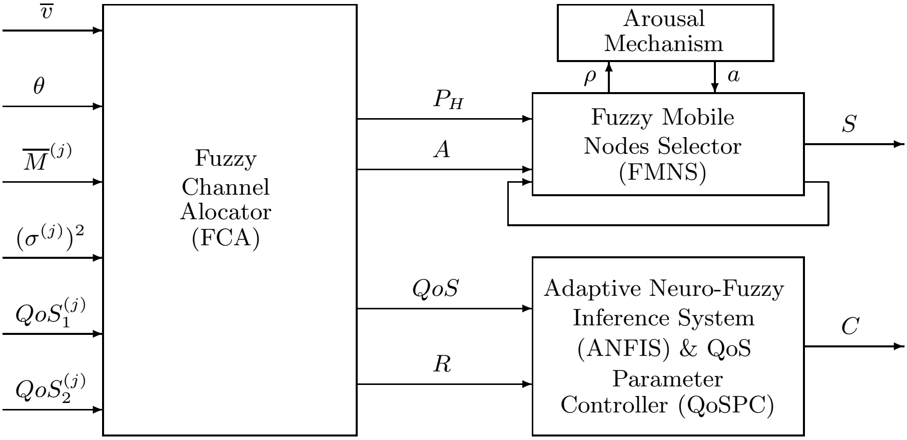

$\overline{v}$
θ
$\overline{M}^{(j)}$
$(\sigma^{(j)})^2$
$QoS_1^{(j)}$
$QoS_2^{(j)}$
Fuzzy Channel Alocator (FCA)
P_H
A
QoS
R
Arousal Mechanism
ρ
a
Fuzzy Mobile Nodes Selector (FMNS)
S
Adaptive Neuro-Fuzzy Inference System (ANFIS) & QoS Parameter Controller (QoSPC)
C

Fig. 2. The block diagram of Fuzzy/Neural Mobility anf Bandwidth Prediction Controller for mobile ad hoc and sensor networks

3 Mobility and Bandwidth Prediction Controller Using ANFIS for Mobile Ad Hoc and Sensor Networks

In this section, we present a mobility and bandwidth prediction system using ANFIS for mobile ad hoc and sensor networks. We assume that our mobility and bandwidth prediction system is implemented in each clusterhead node of the mobile ad hoc and sensor network.

The architecture of our system contains the following functional blocks: Fuzzy Channel Allocator (FCA), Fuzzy Mobile Nodes Selector (FMNS), Adaptive Neuro-Fuzzy Inference System (ANFIS), QoS Parameter Controller (QoSPC). The block diagram is shown in Fig. 2.

In our FCA block, we used as input parameters: v - the speed of mobile nodes, θ - the direction of the movement of mobile nodes, PH - the handoff probability, $\overline{M}^{(j)}$ - the mean rate of j-th mobile node, $(\sigma^{(j)})^2$ - the variance of packet rate, and two Quality of Service (QoS) parameters for jth mobile node, $QoS_1^{(j)}$, $QoS_2^{(j)}$. In our approach the QoS parameters are corresponded to PLR - *packet loss ratio* and to PDV - *packet delay variation*, respectively.

The first two input data are used for predicting the future location of the mobile nodes in the network. The state k at the moment t is given by S_k^t, $k = 1, \ldots, K$, $t \in T$ of the movement of a mobile node. We assumed that $S_k^t = (v_k^t, \theta_k^t)$, where v_k^t is the k-th speed at time t, θ_k^t is the k-th movement direction at time t.

The second two input parameters are needed for prediction the bandwidth of connections. The required bandwidth, C to support all the existing connections with the specified PLR can be given [9] by means of the expression:

$$C = \overline{M} + \xi\sigma \tag{3}$$

where $\overline{M} = \sum_j \overline{M}^{(j)}$, $\sigma^2 = \sum_j (\sigma^{(j)})^2$, the parameter $\xi \cong 1.8 - 0.46 \log_{10}(\eta)$, $\eta = \frac{\overline{M}\sqrt{2}\pi}{\sigma} * \epsilon_{tar}$, ϵ_{tar} is the target PLR.

We supposed that during the elapsed time period τ is a changed movement state. To predict the future movement state S_k^{t+1}, we use the movement function S_k^t, S_k^{t-1}, ..., $S_k^{t-(h-1)}$, where h is the elapsed period time. Then, S_k^t, S_k^{t-1}, ..., $S_k^{t-(h-1)}$ is the location history including the movement patterns. To construct the fuzzy rules from the movement states we used a clustering technique. In this method the cluster center is used as the central value of the fuzzy membership function in the antecedent part of a fuzzy rule. Let $S_k^t = S_k^t, S_k^{t-1}, \dots, S_k^{t-(h-1)}$ be the k-th moment state of the location state k at the moment t and $R_i^{(t)} = R_i^{t,1}, R_i^{t,2}, \dots, R_i^{t,h}$ be the i-th cluster center among M cluster, $i = 1, \dots, M$. Thus, a similarity degree between R_i and S_k^t is given by

$$D_i = \mid S_k^t - R_i^t \mid \tag{4}$$

where D_i^t is the distance at the moment t between R_i and S_k^t. For each cluster i we assumed the radius r_i. If $D_i^t \leq r_i$, then the movement state vector is a cluster.

For the sake of a better description of the movement rules, we used the concept of the *shadow cluster* [5]. According to this concept the shadow cluster defines the area of influence of a mobile node. In other words, a shadow is a set of clusters with the clusterheads to which the mobile node is likely to attracted in the near future. Each clusterhead in the shadow cluster anticipates the mobile's arrival prediction and reserves resources.

Therefore, we introduced rule database associated with the influence of a shadow cluster. We divided our movement state vector into three subvectors. We can describe our database of rules in the form:

$$Rule^j : IF\ (x_j^1\ is\ U_j^1\ AND\ \dots AND\ x_j^q\ is\ U_j^q)\ THEN\ (y\ is\ U_i^p) \tag{5}$$

where U_j^i, $i = 1, \dots, q$, were assumed as the central value c_j^i of the fuzzy membership function defined as follows

$$\mu_{U_j^i}(x_j) = \exp[-\frac{1}{2}(\frac{x_j^i - S_k^{t-(j-1)}}{r})], \quad j = 1, \dots, q,\ i = 1, \dots, c \tag{6}$$

where $U_i^{p,r}$ corresponds to the central value c_j^i and r corresponds to the variance σ_i^j.

The consequences of fuzzy rules are defined here as follows:

1) $U_i^1 = S_k^{t+1}$
2) $U_i^2 = \frac{\sum_k^{m_i} S_k^{t+1}}{m_i}$

The first class of rules in the rule database corresponds to a movement state vector which concerns all the mobile nodes moving outside of the shadow cluster.

The second class of rules describes the change in the shadow cluster which is associated with the direction of movement of the mobile node. This rule allows us to evaluate the future movement within a cluster.

All the fuzzy rules belonging to the second class are additionally identified by a type of arousal, a mechanism that monitors the compatibility of the current beliefs with the observations [13]. This information is translated into an arousal index used to influence the learning process.

Let us denote the arousal index by $a_i \in [0,1]$. The arousal index is updated as follows

$$a_{k+1} = a_k + \beta[(1 - \rho_{k+1}) - a_k] \tag{7}$$

The value of $\beta \in [0,1]$ controls the rate of the change of arousal. In our approach the parameter β is given by

$$\beta_i = 1 - \exp(-\lambda \cdot age_i) \tag{8}$$

where age_i is the age of the i-th fuzzy rule, λ is the parameter which controls the age of a rule.

The young fuzzy rule maps the latest movement state better. The old fuzzy rule is of less significance in the rule database. The closer β is to one, the faster the system is to sense compatibility variations.

The arousal index can be viewed as the complement of the confidence in the belief structure currently held.

An algorithm for mobility learning and prediction uses the participatory learning procedure. The detailed steps of it are as follows:

Input: Let $S_k^t = [S_k^t, S_k^{t-1}, \ldots, S_k^{t-(h-1)}], k = 1, \ldots, K,$ $a \in [0,1]$, $\beta \in [0,1]$, $t = 1, 2, \ldots, T$.

Output: Future movement state S_k^{t+1}.

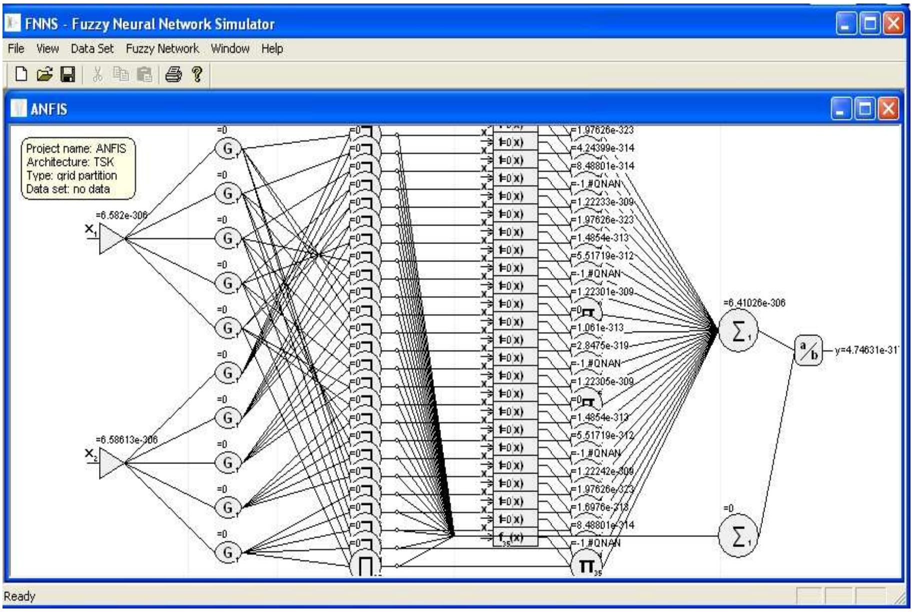
1. For $k = 1, 2, \ldots, K$;

- **1.1** Compute S_1^t and the first future movement state S_1^{t+1}. Establish a cluster center R_1 at S_1^t. Let $U_1 = S_1^{t+1}$. Assign the values to r, ρ_1, age_1.
- **1.2** Determine the kth movement state vector S_k^t for M clusters with centers $R_1, R_2, \ldots, R_M$. If Eq. (4) is satisfied, then the nearest cluster to S_k^t is $R_{k'}$, $k' = 1, 2, \ldots, M$.
  - **a)** If the value of $D_{k'} > r$, create $S_{k'}^t$ as a new cluster. Set $R_{M+1} = S_k^t$, $U_k = S_{k'}^{t+1}$, $age_{M+1} = 0$.
  - **b)** If $D_{k'} \leq r$ then for $l = 1, 2, \ldots, M$ compute $a_{k'}^l$ - arousal index, namely

$$a_{k'}^l = a_{k'}^{l-1} + \beta[(1 - \rho_{k'}^l) - a_{k'}^{l-1}]$$

  Set $age^l = age^l + 1$.

- **1.3** Eliminate one cluster with the lowest arousal index.



Project name: ANFIS
Architecture: TSK
Type: grid partition
Data set: no data

Fig. 3. The connectionist structure of ANFIS controller

1.4 Compute the future movement state S_k^{t+1} at the kth movement state. Compute the parameter U_k.

2. Compute $error = \parallel U_k - U_{k-1} \parallel$.

3. If $error > \epsilon$ then return to step 1; else stop.

3.1 Adapive Neuro-Fuzzy Inference System (ANFIS)

For the correction of the QoS parameters for all connections with mobile nodes that are in use, we applied the ANFIS controller. The term sets of the input linguistic variables are used to describe the QoS change and eventually reject (R) the connection. The output linguistic variable is associated with the bandwidth (C). The connectionist structure of the ANFIS (see Fig. 3) is based on the four-layer Wang-Mendel neural fuzzy network [3], [4]. All the elements of the first layer realized membership functions A_i^j, $i = 1, \ldots, n,\ k = 1, \ldots, N$, where n is the number of controller inputs, N is the number of rules. The second layer consists of the elements corresponding to the rules from the database of our ANFIS controller. In the defuzzification block we used the center-average defuzzification method.

4 Simulation Results

In this section, we evaluated the performance of our mobility and bandwidth prediction controller for the ad hoc and sensor networks through simulation experiments.

The experimental data of the mobility data had approximated to mobile nodes real movement and the real requirement of bandwidth. In our approach, the movement behaviour was defined as the change of location from one given geographical location to another over a two dimensional space. The traveling model can be described as a random-walk mobility model [12] or by the urban areas [8]. With the use of the above method, we have extracted the mobility data in terms of the nodes movement and change of the movement direction.

We have described the speed of a mobile node in the time period $t + 1$ as the random speed which consists of the random speed $\overline{v}_t$ at a given period t and $\tilde{v}_t$ the variation of the mean speed at time period t, namely

$$v_{t+1} = \overline{v}_t + \tilde{v}_t \tag{9}$$

where $\tilde{v}_t$ is selected by the use of a uniform distribution in the speed variation range.

Assuming that a mobile node moves towards its route, we define the movement direction at time period $t + 1$ as

$$\theta_{t+1} = \overline{\theta}_t + \frac{2\pi}{d} \cdot k \tag{10}$$

where $\overline{\theta}_t$ is a current direction towards the given route at the moment t. d is the number of movement directions, k is the movement direction index selected in a random way with the probability P_k, $p_k \in k \in \{0, 1, \ldots, d-1\}$. If all the probabilities are equal, then we obtain the movement towards the destination.

With the use of the above presented mobility model, we have classified the mobile nodes into two categories. The first of them is associated with the mobile sensors which are the movement along an unknown route according to their movement directions. The second one concerns the mobile nodes in an ad hoc network which know the movement path and harvest the data from the clusterhead nodes. In our approach the radius of nodes is equal to 30 meters in the sensor network and 100 meters in an ad hoc network, respectively. Each mobile network has different bandwidth requirements. For example, the mobile nodes in the sensor network need up to 150 kbps. The transmission in the mobile ad hoc network requires some classes of traffic rate. We have admitted two traffic classes, namely CBR and rt-VBR traffic classes with required QoS parameters.

We assumed that the mean speeds are $\overline{v}_t = 5$ km/h for the mobile sensor network and the 40 km/h for ad hoc network, respectively. The speed variation ranges are equal to [10, 25] for the sensor network and [40, 60] for the ad hoc network. Direction probabilities are given in the direction probability vectors $\mathbf{p}$, such that $\sum_{k=1}^{d-1} p_k = 1$.

The performance measure of our controller was determined as the mean number of the realized connection handoffs between the consecutive clusterhead nodes and the mobile nodes. It is defined as the following quotient, namely

$$MS(t) = \frac{A(t)}{R(t) + A(t)} \tag{11}$$

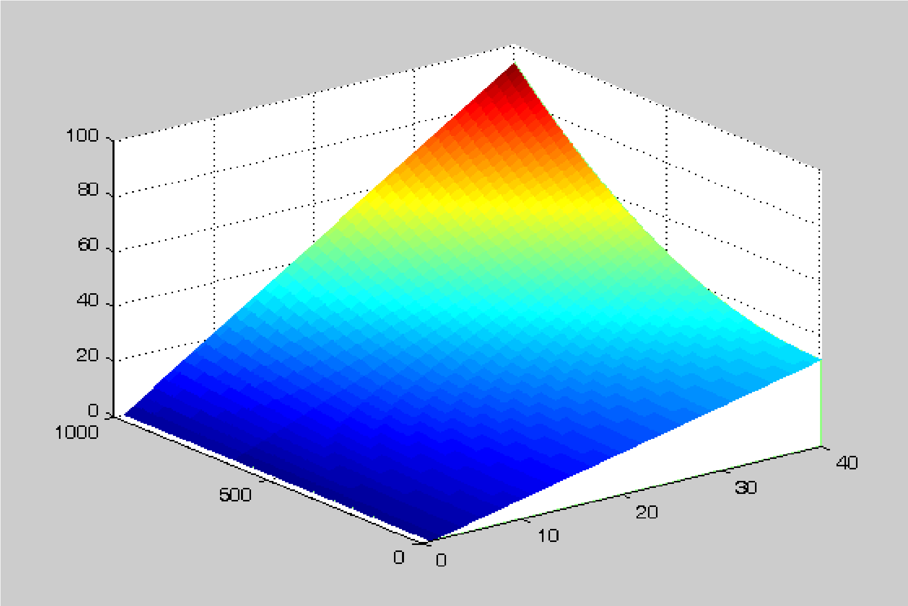


Fig. 4. The percentage of uncorrectly predicted connection handoffs for mobile ad hoc network in dependences of speed of mobile nodes (up to 40 km/h) and required bandwidth (up to 1000 kbps)

where $A(t)$ is the number of the accepted connection handoffs between the consecutive clusterhead nodes and the mobile nodes, $R(t)$ is the number of the rejected connection handoffs between the consecutive clusterheads and the mobile nodes.

The performance of our mobility and bandwidth prediction scheme is determined by the degree of the accuracy of prediction. This parameter depends on such parameters as the speed of mobile nodes and the required bandwidth of the realized transmission. As the measure of the accuracy of prediction we used the percentage of the number of the uncorrectly predicted connection handoffs to the number of all the predicted connection handoffs. This measure is the function of the speed of mobile ad hoc nodes (up to 40 km/h) and the required bandwidth of transmission (up to 1000 kbps). The values of the performance measure for the parameters given above are shown as the surface shown in Fig. 4. We can see that the percentage of uncorrected connections sharply go up at the increasing speed value and required bandwidth. Nevertheless, both traffic classes can be realized without hindrances. But in the case of the fluctuating speed the accuracy of prediction is comparable to the regular speed for the ad hoc network.

5 Conclusions

In this paper, we have proposed the controller based on the ANFIS system for mobile ad hoc and sensor networks. It can predict the future locations of mobile nodes, as concerns the required bandwidth with needed QoS parameters.

Thus, the mobile nodes can maintain a continuous transmission with the clusterhead nodes, regardless of the new location to which they move. Througout the movement of mobile the station the transmission has the same efficiency.

From the simulation experiment of the communication performance the following has been concluded:

1) The controller can replace the complicated electronical devices for switching between clusterhead nodes during the movement of mobile stations if it is learned with the set of the mobility data relevant to the movement patterns.
2) If the transmission bandwidth is constrained, the controller exhibits a better efficiency than for the transmission with a sufficient broadband bandwidth.

Finally, the controller is independent of the architecture of the underlying wireless ad hoc and sensor networks. Therefore, it may be a reasonable technique to consider for the mobility and bandwidth prediction in these networks.

References

1. Basagni, S., Conti, M., Giordano, S., Stojmenović, I. (eds.): Mobile Ad Hoc Networking. IEEE Press/John Wiley & Sons, Piscataway (2004)
2. Hong, D., Rappaport, S.: Traffic Model and Performance Analysis for Cellular Mobile Radio Telephone Systems with Prioritized and Non-prioritized Handoff Procedure. IEEE Trans. on Vehicular Technology 35, 77–91 (1986)
3. Jang, J.S.R.: ANFIS: Adaptive-network-based Fuzzy Interference System. IEEE Trans. on Systems, Man, and Cybernetics 23(3), 665–685 (1993)
4. Jang, J.S.R., Sun, C.T.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice-Hall, Inc., Englewood Cliffs (1997)
5. Levine, D.A., Akyildiz, I.F., Naghshineh, M.: The Shadow Cluster Concept for Resource Allocation and Call Admission in ATM-based Wireless Networks. IEEE/ACM Trans. Networking 6, 1–10 (1997)
6. Liu, G.Y., Maquire Jr., G.Q.: A Predictive Mobility Management Algorithm for Wireless Mobile Computation and Communication. In: Proc. IEEE Int. Conf. Universal Personal Commun., pp. 268–272 (1995)
7. Liu, T., Bahl, P., Chlamtac, I.: Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks. IEEE J. Select. Areas Commun. 16(6), 922–936 (1998)
8. Oppenheim, N.: Urban Travel Modeling. John Wiley and Sons, Chichester (1994)
9. Ren, Q., Ramamurthy, G.: A Real-Time Dynamic Connection Admission Controller Based on Traffic Modeling, Measurement, and Fuzzy Logic Control. IEEE J. on Selected Areas in Comm. 18(2), 184–196 (2000)
10. Shorey, R., Ananda, A., Chan, M.C., Ooi, W.T. (eds.): Mobile, Wireless, Sensor Networks. IEEE Press/John Wiley and Sons Inc., Hoboken (2006)
11. Tabbane, S.: An Alternative Strategy for Location Tracking. IEEE J. Select. Areas Commun. 13(5), 880–892 (1995)
12. Thomas, R., Gilbert, H., Mazziotto, G.: Influence of the Movement of Mobile Station on the Performance of a Radio Cellular Network. In: Proc. of Third Nordic Seminar on Digital Land Mobile Radio Communications (1988)
13. Yager, R.R.: A Model of Participatory Learning. IEEE Trans. on Systems, Man and Cybernetics 20(5), 1229–1234 (1990)

Heuristic-Based Approach for Constructing Hierarchical Knowledge Structures

Fu-Ching Tsai[1], Yi-Chung Cheng[2], Sheng-Tun Li[1], and Ciian-Yuan Chen[1]

[1] Institute of Information Management, National Cheng Kung University,
701, Tainan, Taiwan, R.O.C.
[2] Department of International Business Management, Tainan University of technology,
710, Tainan, Taiwan, R.O.C.
{Fu-Ching Tsai,Yi-Chung Cheng,Sheng-Tun Li,Ciian-Yuan Chen,
r7895104}@mail.ncku.edu.tw

Abstract. Knowledge structure identify the way how people think and provides a macro view of human perception. However, the usability of knowledge is limited due to its structural inconsistency and complexity which makes it difficult to communicate and share. Without knowledge transferring, individuals and organizations are not capable to achieve better performance by learning and communication from others. Previous researches exhibit several disadvantages, such as multiple inheritance and lacking hierarchical features, in state-of-the-art techniques. To tackle this critical matter, we propose a new tree-based knowledge structure for achieving knowledge interoperability and enhancing structural comprehensiveness. According to our evaluation results; the proposed method successfully formulates the hierarchical relations from human cognition and increases the complexity in knowledge navigation and visualization.

Keywords: Knowledge management, concept hierarchy, information retrieval, tree-based knowledge structure.

1 Introduction

In knowledge management, knowledge without interoperability is the main reason impeding the sharing and communication among various applications. For example, how can we merge or compare two experts' knowledge which is represented by network and lattice, respectively? The structural heterogeneity is caused by different input formats, such as rating scales for repertory grid, boolean relation matrix for formal concept analysis (FCA) and numerical similarity table for hierarchical agglomerative clustering (HAC) method etc., thus, the key challenge is to find a universal format as the basis to accommodate converting from various knowledge sources. Moreover, different knowledge representation formats are being broadly discussed along with their advantages and disadvantages in previous researches [1]. The more complex the structures, it is more difficult to visualize its morphology. Unfortunately, the great quantities of elements or links are adopted in formulating human's knowledge, such as lattice and network, which eliminate the effectiveness in knowledge navigation and visualization [2-4].

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 439–448, 2009.

To solve these two problems, we proposed a new algorithm to construct a tree-based knowledge structure (KS) using numerical matrix as input and provide comprehensiveness through concise relations. The input of proposed algorithm is the most common intermediate format for data conversion. Various similarity theories, such as cosine measure, Expected Mutual Information Measure (EMIM), Euclidean distance, etc., are intended to support the transformation from diverse sources to numerical similarity matrix. Thus, with unified data source, knowledge sharing, communication and comparing are capable to proceed based on a single form of KS. Furthermore, the proposed algorithm also avoid the multiple inheritance problem, the ancestor is inherited multiple times via different path, which has been challenged for increased complexity and ambiguity [5]. The remainder of this article is organized as follows. In section 2, we overview various hierarchical KS approaches. In section 3, we introduce the fundamental definitions of this research. The tree-based KS construction algorithm is illustrated in detail in section 4, and in section 5 we discuss the experimental results of our work. Section 6 concludes this paper and gives some future works.

2 Hierarchical Knowledge Structures

Hierarchy is the most straightforward mechanism for human in knowledge modeling with respect to subordinate and superordinate associations. Lattice and tree are two representative hierarchical structures which based on in-depth philosophy background in mainstream of knowledge engineering [6, 7]. The main application of lattice, i.e. formal concept analysis (FCA), is to analyze data and provide well structured information, such as investigate and interpret implicit relations, derive implication rules and facilitate knowledge acquisition, for knowledge workers [8-10]. Although, FCA demonstrates its ability in retrieval task and implicit relation discovery [11], there are two drawbacks of FCA worth to be mentioned. First, many tangled nodes and linkages which are inapplicable for knowledge visualization are produced in lattice in order to map n-dimension data to a two-dimensional space. Furthermore, in practical, lattice structure is too large to display in visual perspective in their entirety. Second, FCA can’t take care of extreme amount of source data[7] due to its high complexity of $O(2^n)$[12].

As to the other hierarchical structure, tree, it is defined as an acyclic graph which is composed of one root node, numerous internal and leaf nodes and edges. In previous researches, tree structure is successfully applied to display massive and complex information by comprehensible visualization and mellifluent navigation. In knowledge management field, the definition of tree is adopted to formulate a layered thematic structure, namely tree-based KS. Nodes and weighted edges represent knowledge objects and their correlations, respectively. There are many methodologies as the way to produce tree-based KS. Minimum Spanning Tree (MST) is often used to retain the most essential correlations in tree construction process. However, MST is not capable to indicate the upper and lower relations while lacking the important start point, the root. Strictly speaking, MST is not a tree-based KS, but more like a condensed network, although it successfully holds the most potent links in a unique structure. Regarding to hierarchical agglomerative clustering (HAC), it is a bottom-up clustering algorithm which treat each node as a singleton followed by iteratively merge most similar pairs until cluster number corresponds to predefined criterion.

In hierarchical KS, multiple inheritance causes the ambiguity for nodes which contain cross-link to different superordinate. For example, shoes for girls are not only ornaments, but also a kind of daily necessaries. In tree structure, there should be two respective links from shoes to its corresponding parent nodes, ornaments and daily necessaries. However, we believe that it is proper to classify each element into only one category in a specific domain to avoid ambiguity. In addition, the complex linkages from multiple inheritances violate the fundamental notion of knowledge representation and impede its recognition and comprehension. Previous researches have paved the way to explore hierarchical feature, nevertheless multiple inheritance problems are not been resolved yet [8, 9, 13-16].

3 Concepts of Hierarchy

3.1 Hierarchical Feature

The definition of tree hierarchy is considered the fundamental principle for node locating and edge growing. We adopt the notion of core ontology from Cimiano et al.[14] to define the tree hierarchy (TH) through Definition 1.

Definition 1 (Tree Hierarchy)

A $TH := (N, root, \leq)$ *consisting of (i) a set of nodes N, (ii) such that* $\forall n \in N$ and $\forall n \leq root$ *(iii) where root is located at the highest level of TH and* $root \in N$

From Definition 1, TH is composed of a set of nodes including a root which locates at the top level of hierarchy. The superordinate and subordinate are formulated by partial order. Based on Definition 1, we can extend partial order relations to depict hierarchical feature in terms of mutual similarities.

Definition 2 (Hierarchical feature)

$$\forall n \in N,\ sim(n,\ ancestor_i(n)) \leq sim(n,\ ancestor_j(n)), \text{if } i \leq j \text{ and } 1 < i \leq j < level(n) \cdot$$

Where $ancestor_i(n)$ represents the upper nodes n which locates at level i and $sim(n,\ ancestor_i(n))$ stands for the similarity between node n and $ancestor_i(n)$. Following the definition of tree structure, we defined the lower the nodes the larger of its level, which means level(*root*), equals to one, is the smallest number of entire hierarchy. The main concept of hierarchy feature is that for any two upper nodes, $ancestor_i(n)$ and $ancestor_j(n)$, of node *n*, their mutual similarities are disproportional with their interval edge numbers. It is worth noting that, by applying the basic nature of hierarchical feature in Definition 2, we don't have to define and select features for each node in order to facilitate the intrinsic hierarchical relations, such as attribute selection in FCA. Instead, the general and specific features are clearly discriminated by comparing mutual similarities.

3.2 Sibling Independence

Siblings in tree structure are defined as nodes which locate in the same layer and share the same parent. Therefore, more general concepts in the parent node are supposed to be divided into several specific parts to take as its child nodes. In tree construction process, sibling independence is a critical factor for determining its width and depth which are the main principles to decide the structure of KS. However, only few researches focused on this important issue. If we choose to apply stronger sibling independence in tree structure, some nodes with too much overlap would alter its location as child nodes instead of siblings because they are not distinguishable enough. More overlap between nodes indicates more similar but loose independence and vice versa. In this research, we proposed a natural criterion of sibling independence to decide the location for each node. That is, the sibling independence of node *n* is designed to compare with its parent, i.e. if similarity between *n* and its sibling is larger than its parent; then *n* should locates in the deeper level instead of current one. Theoretically, siblings with stronger independence are more distinguishable and capable to avoid subsumption effect from one another. However, a straight vertical tree without siblings might be produced if we only consider raising the independence to the limit. On the contrary, hierarchical feature would be diminished with excessive loose sibling independence and derive a flat list-like structure. For the purpose of accurately codify KS from human cognition; we adopted different degrees of sibling independence as the parameter for KS construction. The further analysis of its impact was demonstrated in the later section.

4 Construction of Tree-Based KS

Fig. 1 depicts the three main stages of tree-based KS construction framework which are knowledge codification, similarity refinement and construction algorithm. The detailed procedures are discussed as follows.

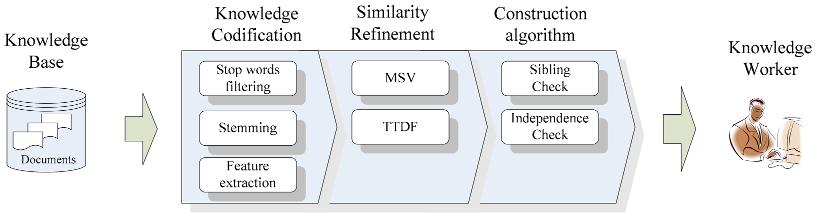


Fig. 1. Three stages of tree-based KS construction

4.1 Knowledge Codification from Corpus

The various perspectives of knowledge are widely embedded in and carried through multiple entities, especially in documents. Undoubtedly, documents are the main source of knowledge. Thus, we used documents to formulate various KOs with specific description to corresponding topics. Since the mutual similarities of KO are adopted for constructing tree structure, traditional information retrieval (IR) techniques are applied in this stage to calculate similarities of documents. First, the data preprocessing process includes tokenization, stop words filtering and stemming. Tokenization represents the division of text into words or terms. Stop words filtering omits words like prepositions and pronouns which are not representative of the documents. Stemming identifies the root form of words by removing suffixes and prefixes. Second, weights of each word is measured by term frequency (TF) and inverse document frequency (IDF) in order to produce the vectors which represent document features. TFIDF is a statistic measure which often used in IR and text mining to evaluate the importance of words in corpus. TF provides the occurrence frequencies of terms appear in a document and IDF is used for distinguishing relevant terms in the corpus.

Finally, mutual similarities of documents are calculated through vector space model (VSM), which is an algebraic model that transforms documents into vectors in a multi-dimensional space. VSM is capable of facilitating the estimation of similarity among documents by calculating the inner product of vectors.

4.2 Similarity Refinement

In most cases, mutual similarities are extracted from representative documents which are typical and relevant to the specific domain. However, synonyms, different co-occurrence patterns and various writhing styles usually affect the similarity measure in IR researches. Thus, we attempt to explore other information from text to refine similarities. The idea is that if we can indentify upper level nodes in advance and connect them in upper part of the structure, then the result will be more reasonable without doubt. We refer the tendency of upper level nodes as abstractness which means the degree of generality of nodes within repository. The abstractness is composed by following measures.

(1) Mutual Similarity Variance (MSV)
As mentioned above, higher level nodes are more general than lower level ones; therefore, the similarity distribution of nodes at higher level should be more diverse than ones at lower level. For this purpose, variance, a statical measure to represent the degree of spread compare with mean, is appropriate to evaluate its abstractness. Therefore, we used MSV as the index which stands for the degree of abstractness to revise the initial similarity for acquiring more reasonable similarities. The MSV for each node is calculated by equation 1.

$$MSV(n_i) = \frac{\sum_{j=1}^{|N|-1}\left(sim(n_i, n_j) - \overline{n_i}\right)^2}{|N|-1}, \text{where } n_i, n_j \in N \text{ and } i \neq j \tag{1}$$

where $\overline{n_i}$ is the average of mutual similarities of node n_i . We ignore the self similarity, always equal to 1, in calculating variance, therefore, the denominator of MSV is $|N|-1$.

(2) TTDF (Top Term Document Frequency)
TTDF is used to estimate document generality of document d_j with cumulative document frequency of topN frequent terms k_i in d_j. TopN frequent terms of a document can be treated as its typical features[17]. Therefore, by summarizing the df of k_i in d_j, we can derive the generality of d_j. Sanderson et al. [16] pointed out that the generality and specify of terms are determined according to its apparent frequency. Therefore, TTDF index is successfully reflect the abstractness of document. TTDF is obtained by using the following equation 2.

$$TTDF(d_j) = \sum_{i=1}^{topN} \sum_{j=1}^{|N|-1} df_{ij}(k_i), \ 1 < j < |N|-1 \text{ and } 1 \le i \le topN \tag{2}$$

Where topN is the number of top ranked terms in documents d_j and $df_{ij}(k_i)$ represents the document frequency of term k_i in document d_j. For considering avoid noise data or enrich information caused by respectively adopting too many top terms and too few top terms in this research, we used top three frequent terms, suggested in Sanderson's work, to represent document feature and calculate their abstractness.

For emphasis the importance of nodes with higher abstractness, we adopt exponential operation in evaluating the degree of abstractness. The abstractness of node n_i is computed as equation 3.

$$abstractness(n_i) = \frac{MSV'(n_i)^x + TTDF'(n_i)^x}{2}, \ x > 1 \tag{3}$$

Where $MSV'(n_i)$ and $TTDF'(n_i)$ ranging from 0 to 1 are the normalized MSV and TTDF, respectively and x is the power of them. Equation 4 depicts the weighted average operation of revised similarity which is averaged with a weight parameter, α, to decide the degree of refinement by abstractness.

$$rev\_sim(n,n') = (1-\alpha)\times sim(n,n') + \alpha \times abstractness(n'), \ 0 \le \alpha \le 1 \tag{4}$$

4.3 Tree Construction Algorithm

The algorithm starts from assigning one node as root, and create its child nodes layer by layer on the basis of mutual similarities and sibling independence. At second layer, the node which is most similar with root is considered as the first child. Similarly, the remanding nodes will connect to root sequentially according to their similarities. However, for avoiding too many related nodes locate in the same layer, sibling independence criterion is applied to eliminate the overlap among siblings. In next step, we need to classify each unlinked node to one of its parent which is the most similar with it. For each group, the same procedures, i.e. sibling independence and hierarchical feature, are invoked recursively until each node has been linked to the KS. However, for some circumstances, not all nodes are able to be connected because of the violation of hierarchical feature. The reasonable explanation is that the incomplete tree structures are not appropriate to extract inherent hierarchies from knowledge repositories. Therefore, we ignore the incomplete trees in this research without future discussions and the incomplete trees will not be displayed for knowledge workers.

5 Evaluations and Discussions

5.1 Data Collection

The public data of Association for Computing Machinery (ACM) Computing Classification System (CCS) are applied as golden standards to demonstrate the consistency with produced tree structure. ACM CSS is an acyclic tree based categorization scheme without multiple inheritances, which adapt to our definition of tree structure, to accurately reflect the essential structure of the fields of computer science and information systems. Among various categories (from A to K), we choose B (hardware) and I (computing methodologies) as corpus of our research due to more abundant subcategories compare with others. 20 document abstracts in each subcategory are randomly selected to represent its knowledge content. However, several subcategories with less than 20 abstracts are eliminated in this research to avoid the unreliable similarities caused by insufficient data. Table 1 depicts the statistics of two golden stands in this research.

Table 1. The statistics information of ACM gold stands

Golden standard	Nodes	Height	Documents	Terms
ACM CCS, category I	51	3	1020 abstracts	154,148
ACM CCS, category B	37	3	740 abstracts	113,997

5.2 Evaluations

Since the tree-based KS construction is affected by various parameters, i.e. similarities of KO, degree of refinement, sibling independence, which makes the experimental process so tedious in determining the ranges of parameters. To better understand the adequacy and effects of parameters, an experimental design process is applied in this research to decrease the experimental cost and determine the optimal parameters to achieve better recommendation performance. The proposed algorithm involves four parameters: TFIDF, degree of similarity refinement, sibling independence and power of exponential operation. The criteria of TFIDF influence the feature numbers that forms a basis as making relevance decisions [18]. Higher TFIDF successfully avoid the impact from noise terms; likewise some information might lose because of fewer features are kept in document vectors. Different level of power is designed to exhibit various degrees of importance of general nodes. Likewise, the weight of similarity refinement demonstrates the various degree of influence on similarity adjustment. In order to investigate the structural impacts of KS, different values of sibling independence are tested for forcedly modifying the width and depth of KS.

The performance evaluations indicate the relative importance of four parameters and provide the optimal combination of parameters in two gold standards. To validate the performance of this research, we adopt taxonomy overlap (TO) [14] to demonstrate the consistency of golden standards and produced structures. In the aspect of structure comparison, there are various measures proposed in the previous researches [19, 20]. However, TO provide a more general feature comparing with other metrics and emphasize the importance of upper level nodes which fits the inherent characteristic of KS.

We follow the combination of optimal parameters to investigate TO performance in this research. Table 2 indicated that TO in ACM CCS I and B are 72.41% and 72.90%, respectively. The TO results represent a great achievement of proposed tree-based structure as the relevance of hierarchical feature within knowledge repositories. In addition, a single universal set of parameters for two data sets is essential to be further discussed. We choose the parameter combinations with the least accuracy loss as the universal set. Table 2 reveals that all parameter settings are identical in these two data sets, except similarity refinement, which are 0.2 in I category and 0.5 in B category. We select the second best setting of similarity refinement in ACM CCS category B, which is 0.2, to perform further experiments. The experiments with universal parameter settings show that TO for two data sets are 72.41% and 71.93%, respectively. Accordingly, the results represent the robust parameter setting for TO performance of all data sets and efficient for providing knowledge user hierarchical KS with high accuracy.

Table 2. The optimal set of parameter in TO evaluation

Gold standard	TFIDF	Power	Similarity refinement	Sibling independence	TO
ACM CSS I	0.02	2	0.2	0.9	72.41%
ACM CSS B	0.02	2	0.5	0.9	72.90%

6 Conclusions

In the era of information overloading, knowledge representation technique is important to cognitively demonstrate tacit knowledge. In previous researches multiple inheritance and absence of root are regarded as the main weaknesses in providing clear and comprehensive KS. In order to overcome these issues, we proposed a methodology to construct a tree-based KS which is single inherited and provide a root to enhance the navigation of knowledge. In order to justify the effectiveness of proposed methodology, two public databases were adopted as gold standards to evaluate its consistency with proposed tree structures. Experimental results demonstrate the impressive accuracy in ACM CCS database. Moreover, the high accuracy also shows that our methodology is able to reflect hierarchical relation from a flat set of mutual similarities automatically via tree construction algorithm.

Other future research directions may include exploring more effective measure for hierarchical quality evaluation, trying other similarity measure for obtaining mutual similarity matrix and achieving better performance on accuracy. Certainly, experiments for more gold standards are needed to validate the solidity of universal parameter setting.

References

1. Eppler, M.J., Burkhard, R.A.: Visual representations in knowledge management: framework and cases. Journal of Knowledge Management 11, 112–122 (2007)
2. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Computing iceberg concept lattices with TITANIC. Data & Knowledge Engineering 42, 189–222 (2002)
3. Rajapakse, R.K., Denham, M.: Text retrieval with more realistic concept matching and reinforcement learning. Information Processing & Management 42, 1260–1275 (2006)
4. Belohlavek, R., Dvorak, J., Outrata, J.: Fast factorization by similarity in formal concept analysis of data with fuzzy attributes. Journal of Computer and System Sciences 73 (2007)
5. Truyen, E., Joosen, W., Jorgensen, B., Verbaeten, P.: A Generalization and Solution to the Common Ancestor Dilemma Problem in Delegation-Based Object Systems. In: Proceedings of the 2004 Dynamic Aspects Workshop, Lancaster, England, pp. 103–119 (2004)
6. Stumme, G.: Off to new shores: conceptual knowledge discovery and processing. International Journal Human-Computer Studies 59, 287–325 (2003)
7. Priss, U.: Formal Concept Analysis in Information Science. In: Cronin, B. (ed.) Annual Review of Information Science and Technology (ARIST), vol. 40, pp. 521–543. Information Today Medford, New Jersey (2006)
8. Tho, Q.T., Hui, S.C., Fong, C.T.H.: Automatic Fuzzy Ontology Generation for Semantic Web. IEEE Transactions on Knowledge and Data Engineering 18, 842–856 (2006)
9. Chi, Y.-L.: Elicitation synergy of extracting conceptual tags and hierarchies. Expert Systems With Applications 32, 349–357 (2007)
10. Formica, A., Missikoff, M.: Inheritance Processing and Conflicts in Structural Generalization Hierarchies. ACM Computing Surveys 36, 263–290 (2004)
11. Carpineto, C., Romano, G.: Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO. Journal of Universal Computer Science 10, 985–1013 (2004)
12. Ganter, B., Wille, R.: Formal concept analysis: Mathematical foundations. Springer, New York (1999)
13. Chung, W., Chen, H., Nunamaker Jr., J.F.: A Visual Framework for Knowledge Discovery on the Web: An Empirical Study of Business Intelligence Exploration. Journal of Management Information Systems 21, 57–84 (2005)
14. Cimiano, P., Hotho, A., Staab, S.: Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. Journal of Articial Intelligence Research 24, 305–339 (2005)
15. Lammari, N., Metais, E.: Building and maintaining ontologies: a set of algorithms. Data & Knowledge Engineering 48, 155–176 (2004)
16. Sanderson, M., Lawrie, D.: Build, Testing and Applying Concept Hierarchies. Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval, vol. 7, pp. 235–266. Springer, US (2000)
17. Glover, E., Pennock, D.M., Lawrence, S., Krovetz, R.: Inferring Hierarchical Descriptions. In: Proceedings of the 20th International Conference on Information and Knowledge Management (CIKM), pp. 507–514. ACM, McLean (2002)

18. Wu, H.C., Luk, R.W.P., Wong, K.F., Kwok, K.L.: Interpreting TF-IDF term weights as making relevance decisions. ACM Trans. Inf. Syst. 26, 1–37 (2008)
19. Rodríguez, M.A., Egenhofer, M.J.: Determining Semantic Similarity among Entity Classes from Different Ontologies. IEEE Transactions on Knowledge and Data Engineering 15, 442–456 (2003)
20. Tang, J., Li, J., Liang, B., Huang, X., Li, Y., Wang, K.: Using Bayesian decision for ontology mapping. Web Semantics: Science, Services and Agents on the World Wide Web 4, 243–262 (2006)

An Ontology-Based Condition Analyzer for Fault Classification on Railway Vehicles*

Cristina De Ambrosi[1], Cristiano Ghersi[1,2], and Armando Tacchella[1]

[1] Università degli Studi di Genova - DIST - Viale Causa 15, 16145 Genova, Italy
<name>.<surname>@unige.it
[2] Medservice.com s.r.l. - C.so Montegrappa 1/1A, 16137 Genova, Italy
<name>.<surname>@medservice.com

Abstract. In this paper we summarize the design and the implementation of an ontology-based condition analyzer which performs on-board fault detection and classification for railway rolling stock. Our system is engineered to run on embedded devices that meet stringent railway regulations for on-board deployment and thus it witnesses the feasibility of ontology-based approaches for the implementation of intelligent information systems even in contexts where memory and processing resources come at a premium.

Keywords: Ontology-driven information systems design, Knowledge management, Domain-specific ontologies.

1 Introduction

In this paper we study the problem of engineering an ontology-based condition analyzer (CA) prototype which can perform on-board intelligent *fault detection*, i.e., rising alarms when characteristic features exceed given thresholds, and *fault classification*, i.e., assessing the severity of faults. Considering the state of the art in railway transportation (see, e.g., [1]), we see that fault detection is routinely performed by on-board devices – mostly naïve switching-logic programs – while fault classification is most often performed off-board by human experts. The purpose of our work is thus to demonstrate the feasibility of an ontology-based approach to increase the current automation level in the fault diagnosis process.

The CA herein described analyzes the conditions of some critical subsystems that are installed on Trenitalia E414 locomotives, but our approach can be extended to encompass all the subsystems installed on the vehicle, and it can be applied to different vehicles as well, including coaches. At the hardware level, the CA is interfaced with the on-board field bus and collects signals from the vehicle control logic such as traction effort, axle temperatures, battery voltage and current. At the software level, the CA leverages an ontology model that describes (i) the structure of the vehicle, (ii) the context in which the vehicle is operated and (iii) the relationships between observed data, symptoms and faults. The ontology model is loaded in a triple store which is populated

* This work has been partially supported by a research contract from Bombardier Transportation Italy. The authors wish to thank Medservice.com for the support provided and, particularly, Lorenzo Nana and Marco De Luca for their collaboration in the implementation of the project.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 449–458, 2009.

with individuals representing specific measures, as well as observations, symptoms and faults by specialized software agents. Fault individuals are classified using a Description Logic (DL) reasoner. To the best of our knowledge, this is the first attempt to engineer a device performing on-board fault classification using ontology-based automated reasoning. This task is challenging because of two competing needs: (i) railway regulations such as, e.g., EN50155 [2], require hardware which is highly reliable, and whose performances are thus necessarily far even from modest-budgeted desktop PCs; (ii) ontology-related tools, e.g., DL reasoners, have relatively large memory and storage footprints, as well as substantial processor requirements.

The paper is structured as follows. In Section 2, we briefly describe the E414 subsystems and the faults that we need to detect and classify. In Section 3, we give an overview of the ontology design, including the E414 subsystems and the signals of interest, while in Section 4 we outline the relevant details of the CA architecture and its implementation. We provide some experimental results in Section 5, where we consider simulated test scenarios on which to test our prototype, and we conclude the paper in Section 6 with a summary of the current results and our future research agenda.

2 Preliminaries

The CA prototype herein described focuses on four subsystems of the E414 locomotive, namely the traction groups, the braking rheostats, the axle boxes and the battery package. The temperature of each traction group (average temperature of two motors), of the braking rheostats, and of the axle boxes are available as process data from the field bus (Multifunction Vehicle Bus, MVB), as well as voltage and current of the battery package. Considering the signals above, the CA can detect five different faults related to the subsystems under observation, namely high temperatures in the traction groups, braking rheostats and axle boxes, high temperature gradients in the braking rheostat, and low voltage levels in the battery package. The main task of the CA is to perform fault classification according to

- priority for maintenance (high or low), and
- impact on mission-related and safety-related aspects.

In the second case the CA should distinguish between mission-related faults, i.e., faults that may cause a loss in the efficiency of the locomotive and thus hinder the train mission, and safety-related faults, i.e., mission-related faults that may also impact on passengers' or operators' safety. Clearly, not all faults are priority for maintenance, and not all faults classify as mission or safety related. As an example, faults associated to the braking rheostat are always a priority for maintenance since high temperatures may indicate insufficient ventilation of the rheostat, but they are never mission related since electrical braking is not strictly necessary. On the other hand, high temperatures in the traction groups configure both a priority for maintenance and a mission concern, since automatic protection mechanisms cut power whenever temperature rises above a given threshold, and this may cause efficiency loss of up to 100%. In the following, we will refer to traction groups as an example of system that can generate a faulty condition.

3 Ontology Design

From a design perspective, our work builds on the ontologies developed in the context of the Integrail project [3]. In particular, our ontology for the E414 locomotive builds on the SP3A core ontology [4] which in turn, builds on the SOUPA ontology [5]. Briefly focusing on the SP3A ontology, the concepts and properties of interest to us are:

OBSERVATIONDATA is the class of all process variables, i.e., data that we read from the field bus to which numerical or logical values are associated.

OBSERVATION is the class of "interesting" patterns of observations, i.e., sequences of observation data from which individuals of class SYMPTOM and FAULT arise. It is a necessary condition for OBSERVATION individuals to be related to SYMPTOM individuals (HASSYMPTOM property), to TARGETSYSTEM individuals (REFERSTOSYSTEM property), and to a specific OBSERVER (ISOBSERVATIONFROM property).

SYMPTOM is the key concept bridging OBSERVATION and FAULT individuals. Indeed, it is a necessary and sufficient condition for SYMPTOM individuals to be related to OBSERVATION individuals via the REFERSTOOBSERVATION property. In other words, symptoms are defined to be individuals related to interesting pattern of observations. It is also a necessary condition for SYMPTOM individuals to be related to FAULT individuals via the REFERSTOFAULT property. Finally, a SYMPTOM is necessarily associated to an EVENTLEVEL individual (one of the partition HIGH, MEDIUM and LOW) which, in turn, supports the maintenance priority level of the corresponding fault.

FAULT is a concept whose individuals are defined in terms of a necessary relationship with SYMPTOM individuals, i.e., a FAULT individual is necessarily related to some individuals of the class SYMPTOM (via the HASSYMPTOM property). SP3A ontology defines two subclasses of FAULT: PRIORITYFAULT and NONPRIORITYFAULT. The first subclass is meant to contain all the faults that have high priority for maintenance purposes, while the second subclass contains all the remaining faults. Specific faults are not asserted to be low or high priority, but a reasoner can decide where to classify a given fault looking at the event level associated to the symptom.

The SP3A ontology is meant to provide a general model without early-binding assumptions, but we require a specialized ontology to support reasoning about Trenitalia E414 locomotive. As an example of the specialization that we require, in Figure 1 we show the portion of the E414 ontology that deals with traction faults[1]. Looking at Figure 1, we can see that the concept OBSERVATIONDATA has been specialized in subclasses whose individuals correspond to the actual temperature and traction effort samples extracted from the field bus. As for OBSERVATION we can see that TRACTIONHIGHTEMPERATUREOBSERVATION subclass is added.

Still looking at Figure 1, we can see that the specialized subclasses for symptoms and faults have been added. In particular, fault and symptom classification can be obtained by a Description Logic reasoner – Pellet [7] in our implementation. This is done

[1] A detailed description of the ontology model and related pictures can be found at www.mind-lab.it/ConditionAnalyzerE414; the complete ontology, together with supporting ones, is available on-line in OWL/RDF format from [6].

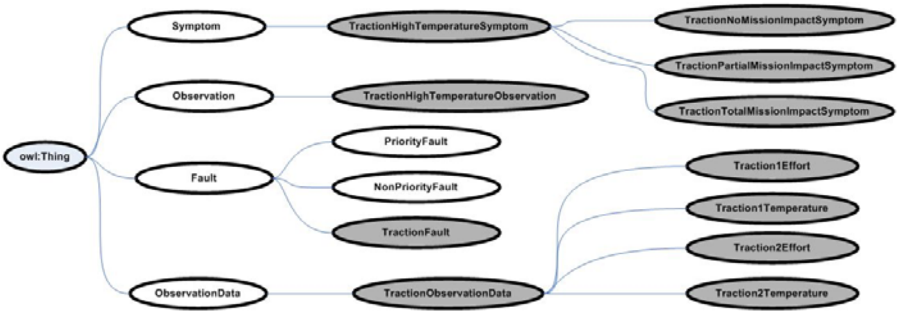


Fig. 1. A portion of the E414 ontology regarding traction faults, represented as a graph. White nodes are SP3A concepts, grey ones are E414-specific concepts.

by considering some properties associated to symptoms and observations and then assessing whether the corresponding fault or symptom is safety-related, mission-related or generic. A similar deduction mechanism, which is predefined in SP3A-ontology, allows the reasoner to infer whether a fault is priority for maintenance or not (concepts PriorityFault and NonPriorityFault).

In the E414 ontology, the classification of faults as mission/safety related is strictly dependent on the classification of the related symptoms. For a fault to be classified respectively as MissionRelated or as SafetyRelated, a necessary and sufficient condition is to be caused by a MissionRelatedSymptom or SafetyRelated Symptom. The classification of symptoms is obtained by reasoning on Observation individuals considering the characteristics of the – possibly anomalous – patterns observed. For instance, in the case ofTractionHighTemperatureObservation, according to expert's advice, we have defined three ranges of temperature that correspond to "interesting" patterns: from 70 to 80 degrees, from 80 to 130 degrees, and over 130 degrees. The E414 ontology postulates that observations falling in the second and in the third ranges are to be considered mission critical, while the ones in the first category are only maintenance critical. According to this logic, the DL reasoner is then able to detect and classify faults starting from the characterization of the underlying observations.

4 Architecture and Implementation

The functional architecture of the condition analyzer is presented in Figure 2. The architecture consists of two groups of components, plus a semantic database engine (the Triple store). The first group performs pre-processing of the data and it is comprised of MVB Adapter, Symptom Discoverer and State Analyzer. The task of these components is to convert the raw data coming from the on-board subsystems in semantically enriched information which is then recorded in the Triple store. The second group performs post-processing of the data and it is comprised of Reasoner and Query Agent. The task of these components is to reason about semantically enriched information in order to infer the classification of symptoms and faults. Notice that in our implementation, the Reasoner is an off-the-shelf tool, while the remaining components have

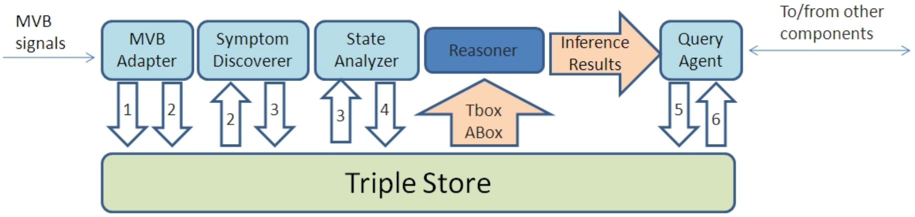


Fig. 2. Functional architecture of the condition analyzer. Each block is a separate functionality, and arrows denote the data flow between blocks: ObservationData individuals (1), Observation individuals (2), Symptom individuals (3), Fault individuals (4), Queries (5), Query results (6).

been designed and implemented from scratch. In the following, we describe each of the components in Figure 2 in some detail.

MVB Adapter. This component interfaces with the on board field bus (MVB in the E414 locomotive), and it parses the incoming signals into an internal format for further processing. Parsed data are saved into a buffer which keeps the history of the latest incoming signals, and it is inspected for anomalous patterns. When such a pattern is detected, individuals of the classes OBSERVATIONDATA and OBSERVATION are created and recorded into Triple Store along with all the properties required by the ontology specification. Whenever this happens, Symptom Discoverer is triggered to analyze the new information. Figure 3 shows an example of a set of samples representing temperatures of a traction group. When the temperature of the traction group is higher than 70 degrees for at least 3 consecutive samples, the adapter starts to keep track of the samples. The anomalous pattern ends either when the temperature goes below 70 degrees or when more than 60 consecutive temperature samples are above the threshold, whichever comes first. Once such a pattern is over, the corresponding individuals in the classes TRACTIONOBSERVATIONDATA (the actual sample values) and TRACTIONHIGHTEMPERATUREOBSERVATION (the anomalous pattern) are recorded into TripleStore.

Symptom Discoverer. This component builds SYMPTOM individuals for OBSERVATION individuals which are not yet related to any SYMPTOM individual. In order to accomplish this task, a SPARQL [8] query is run on the triple store, and the results of such query are used for the creation of new SYMPTOM individuals, along with all the properties required by the ontology specification. Notice that Symptom Discoverer builds generic SYMPTOM individuals, leaving their classification to the reasoner called by Query Agent. For example, if an observation of the class TRACTIONHIGHTEMPERATUREOBSERVATION has been created by MVB Adapter, then Symptom Discoverer detects it and it builds a new SYMPTOM individual. In particular, a triple stating that a specific individual of the TRACTIONHIGHTEMPERATUREOBSERVATION class is related to a new individual of class SYMPTOM by the REFERSTOOBSERVATION property is added to Triple Store.

State Analyzer. This component builds FAULT individuals for each SYMPTOM individual created by Symptom Discoverer, together with the CAUSEDBYSYMPTOM

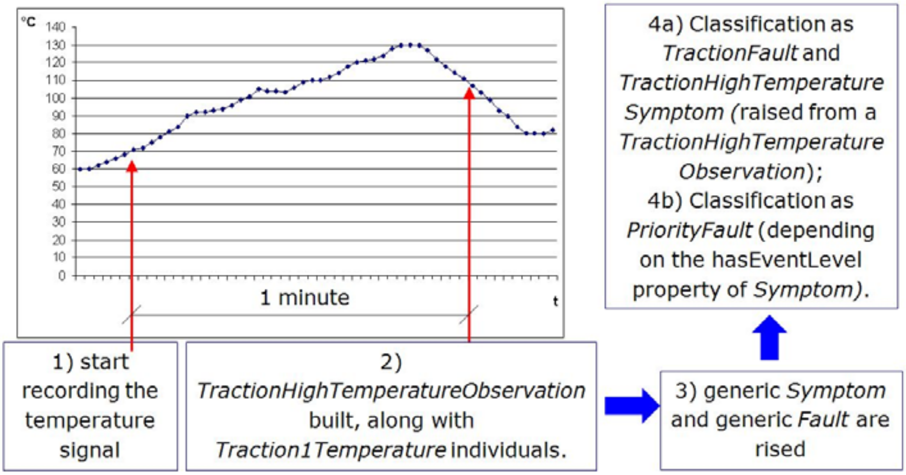


Fig. 3. Temporal flow of a signal tracking Traction Temperature, together with the phases of recognition of incipient fault

property in order to link the related SYMPTOM individual. Notice that also State Analyzer builds generic FAULT individuals, leaving their classification to the reasoner. For example, if a SYMPTOM individual has been created by Symptom Discoverer, then State Analyzer detects it and it builds a new FAULT individual. In particular, a triple stating that a new individual of class FAULT is related to a specific individual of class SYMPTOM by the CAUSEDBYSYMPTOM property is added to Triple Store.

Query Agent. This component has the purpose of executing queries on the triple store and publishing the results, transmitting them to external agents – in the Integrail project [3], a CA on ground that analyzes the condition of a whole fleet, rather than a single vehicle. Queries are run with the aid of Reasoner in order to classify individuals from the existing information. As an example, a query can be executed in order to retrieve all the chains FAULT-SYMPTOM-OBSERVATION-OBSERVATIONDATA together with all the related properties, with the goal of having the whole history of a possible fault condition. During this activity, FAULT and SYMPTOM individuals must be classified to infer their priority for maintenance, their mission or safety related impact and the target system of the (incipient) fault. Let us consider again traction groups as an example. As we have seen above, it is possible to chain back from a given FAULT individual to the specific TRACTIONHIGHTEMPERATUREOBSERVATION individual that represents the corresponding anomalous pattern. As such, the individual is tagged by a distinctive property which determines its severity. For instance, we can assume that:

- o is an individual of the class TRACTIONHIGHTEMPERATUREOBSERVATION whose property ISAT is set to the constant _130DEGREES,
- s is the SYMPTOM individual related to o, and
- f is the FAULT individual related to s.

Table 1. Comparison between Pellet and Racer Pro on an ontology classification task

Number of chains	Pellet CPU Time [s]	Racer Pro CPU Time [s]
1	0.31	1.33
2	0.34	3.80
4	0.47	9.95
8	0.67	38.17
16	1.89	221.89

The E414 ontology postulates that, given an observation which is (i) instance of TractionHighTemperatureObservation, and (ii) related by isAt to the constant_ 130degrees, the corresponding symptom is also an instance of TractionTotal MissionImpactSymptom, which is a subclass of MissionRelatedSymptom. Therefore, the Reasoner can infer that s belongs to the subclass MissionRelatedSymptom because of its relationship with o and the properties of o. Furthermore, E414 ontology postulates that a few subclasses of MissionRelatedSymptom – including TractionTotalMissionImpactSymptom – are able to generate a MissionRelatedFault – a subclass of Fault. Owing to this, f is classified by Reasoner as a mission-related fault, because of its relationship with s.

5 Experimental Setup and Results

In this section we describe three sets of experiments. In the first set, we compare two off-the-shelf reasoners which are most suited for our purposes. In the second set, we test our CA prototype in a realistic environment. In the third set, we investigate more thoroughly the performances of the reasoners with respect to the size of the ontology, both in terms of the number of concepts and properties (model complexity) and in terms of the number of individuals (data complexity). For all the experiments, we have generated use cases scenarios for the E414 locomotive, i.e., a set of multidimensional time series corresponding to the various process variables that the CA reads from the vehicle bus (52 values). The scenarios mimic the real behavior of the signals on the locomotive with randomly injected faults.

As for the reasoners comparison, we investigate the performances of Pellet [7] and Racer Pro [9]. In order to compare them, we used the CA prototype to generate OWL files containing 1, 2, 4, 8 and 16 complete chains of Fault-Symptom-Observation-ObservationData individuals. Such files describe the entire ontology model of the system at a given time, i.e., exactly after the detection of 1, 2, 4, 8 and 16 faults, respectively. The OWL files have been loaded into Protégé [10] – a visual tool for ontology editing and processing – since it is able to communicate with reasoners exposing a DIG-compliant interface in order to classify individuals. This set of experiments run on a standard 1.6 Ghz Intel Celeron M CPU, running Windows XP Pro with 1.5 GB of RAM. As we can see in Table 1, Racer Pro performs worse than Pellet in all the test cases, and the gap between the two reasoners reaches two orders of magnitude for the largest test case. Similar results are also known in the literature (see, e.g. [11]), and this evidence convinced us to drop Racer Pro in favor of Pellet.

Table 2. Results of experiments obtained with two versions of the software: (a) lazy, all the individuals are kept; (b) eager, individuals are dropped as soon as possible

Scenario	Memory Consumption [MB]	CPU Time [ms]	Amortized CPU Time [ms]
1a	38	90	ND
2a	74	25373	25373
3a	106	1053656	210731
4a	OUT OF MEMORY	3253637	191390
1b	37	90	ND
2b	72	21506	21506
3b	104	86938	17387
4b	105	279523	16442

In the second set of experiments, our goal is to ensure that our implementation fits the constraints. To this end, we ran several tests using different fault scenarios. Here we show the results obtained with 4 scenarios, of which the first includes no fault, the second includes only one fault, the third includes five contemporary faults, and the last 17 contemporary faults (maximum number of different faults). All the scenarios last for 3600 samples with no stops, simulating chunks of 1 hour journeys assuming 1Hz sampling of the signals. In order to account for noise, we ran each test several times, and for each statistics we report its maximum value across the runs. To run the simulations we have used two identical embedded PCs in an off-board installation. The two PCs are EN50155-compliant embedded devices featuring a passive backplane architecture and 1GHz Socket 370 FC-PGA Celeron CPU with 256MB of main memory and a 1GB SSD providing mass storage. The operating system is Linux Blue Cat (kernel 2.6) and all the software modules run on a Sun Java Virtual Machine implementation (JRE 1.6).

Table 2 shows the results obtained by running the CA on the scenarios using two different configurations. Configuration (a) is "lazy", i.e., it keeps all the individuals in the triple store, while configuration (b) is "eager", i.e., it deletes individuals from the triple store as soon as possible. The eager behavior is practically feasible because once a given chain OBSERVATIONDATA-OBSERVATION-SYMPTOM-FAULT has been classified and the corresponding information is sent to ground, there is no need to keep it on board, as it will not be subject to further processing. As we can see in Table 2, the eager version results in a great improvement over the lazy one, both in terms of memory consumption and in terms of computation time. In particular, in the second column of Table 2 we can notice that the eager version performs reasonably well, even in the fourth test case (worst-case scenario). In the same scenario, on the other hand, the lazy version exceeds the amount of available memory.

Even if reasoning is a time-consuming task, and we experienced a substantial growth of CPU-time when the number of individuals populating the ontology increases, the computation time of the eager version grows acceptably. Indeed, as we can see in the rightmost column of Table 2, the amortized computation time over a single scenario, i.e. the total time divided by the number of observations, decreases with the number of concurrent observations detected in the round. Eventually, managing a round of samples without detected observations takes only 90 ms, which leaves enough time for other activities, and allows the CA to process all the incoming MVB signals in due course.

From the above results, it seems that the number of individuals in the ontology (data complexity) is crucial for the performances of the reasoner. In order to make sure that

Table 3. Experimenting with different ontologies: (c) *full* E414 ontology (including SP3A and SOUPA); (d) *lean* E414 ontology (including only part of SP3A)

Scenario	Memory Consumption [MB]	CPU Time [ms]	Amortized CPU Time [ms]
1c	37	90	ND
2c	72	21506	21506
3c	104	86938	17388
4c	105	279523	16442
1d	36	90	ND
2d	56	17141	17141
3d	72	65159	13032
4d	72	243971	14351

Table 4. Ontology classification with Pellet and FaCT++ on *full*, *lean* and *core* E414 ontologies. Times are in CPU seconds. ERR: error in DIG communication; MEM: out of memory.

No. of chains	Pellet - full	Pellet - lean	Pellet - core	FaCT++ - full	FaCT++ - lean	FaCT++ - core
1	2.28	0.31	0.05	0.21	0.06	0.01
2	2.31	0.34	0.06	0.22	0.08	0.01
4	2.78	0.47	0.08	0.31	0.10	0.01
8	3.39	0.67	0.14	0.57	0.22	0.06
16	3.59	1.89	0.24	1.24	0.56	0.10
32	7.83	5.94	0.42	3.25	2.05	0.32
64	18.91	16.59	1.02	10.73	5.69	1.09
128	93.80	57.14	4.05	39.34	35.07	4.44
256	> 4hrs	ERR	12.53	239.79	174.42	17.43
512	–	–	56.05	–	678.04	72.98
1024	–	–	222.06	–	–	428.33
2048	–	–	MEM	–	–	1865.20

this is indeed the problem, we reduced the model complexity by building a *lean* version of the ontology – as opposed to the *full* one used so far. In particular, the *lean* ontology does not contain classes, properties and individuals which are not strictly necessary for our purposes. In Table 3 we present the results of the eager version of the CA running on the same scenarios used in Table 2, but considering either the *full* (c), or the *lean* (d) model. Notice that (c) results in Table 3 are the same as (b) results in Table 2. From this experiment, we see that also model complexity is a key factor for the reasoning efficiency, because the run times on the *lean* model scale better than on the *full* one.

In the last set of experiments, we set out to close the issue about data vs. model complexity. In this set, we generate OWL files with up to 2048 observations – and related individuals, when available in the model. We also build a *core* version of the E414 ontology, which considers only Symptom and Observation classes, i.e., the core of our fault-classification. In this experient, we considered also another reasoner, FaCT++ [12]. Indeed, while using FaCT++ in the CA is troublesome because of the DIG interface's limits, using it off-line provides an useful benchmark for Pellet, and ensures that Pellet is not the culprit of the problems experienced. In this set of experiments we used a 3.0 Ghz Intel Pentium PC with 1GB of memory running Linux Ubuntu 8.04.

As we can see in Table 4, both FaCT++ and Pellet have the same behavior. The results on *lean* and, particularly, on the *core* model witness that model complexity is the cause of the scaling issue experienced in the previous sets of experiments. The reason why the concepts and properties that are in the *full* model – but not in the *lean* and *core* ones – can cause such a bad performace is yet to be fully understood. At the moment, we conjecture that the cause could be the massive import of useless concepts from foreign ontologies, due to the overall InteGRail design constraints.

6 Conclusions and Future Work

In this paper we have presented the first attempt to engineer an ontology based information system that runs within the CPU and memory constraints of an embedded computer suitable for rolling stock deployment. Our prototype shows the potential as well as the limits of ontology-based diagnosis for practical and demanding applications. On the one hand, we have succeeded in demonstrating effective on-board intelligent fault classification. On the other hand, we have found that the resources consumed by the reasoner can severely impact on the performances of the whole system, and that such limitations arise when the model complexity is relatively high even if a relatively small number of individuals has to be dealt with. While our current implementation succeeds in bypassing the problem without becoming ineffective, our future work will be focused on making the approach more scalable by trying to reduce the model complexity without losing compatibility with other elements of the system (e.g., the CA on ground).

References

1. The 4th IET International Conference on Railway Condition Monitoring 2008, http://conferences.theiet.org/rcm08/index.htm
2. Railway applications-Electronic equipment used on rolling stock. Standard EN50155 (2005)
3. Intelligent Integration of Railway Systems-FP6 EU project, http://www.integrail.info
4. SP3A ontology, http://www.integrail.info/ont/SP3A.owl
5. Chen, H., Perich, F., Finin, T., Joshi, A.: SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In: 1st Annual International Conference on mobile and Ubiquitous Systems: Networking and Services, pp. 258–267. IEEE Press, Los Alamitos (2004)
6. An ontology for the E414 locomotive, http://www.integrail.info/ont/E414LocoOntology.owl
7. Sirin, E., Parsia, B., Cuenca Grau, B., Kalyanpur, A., Katz, Y.: Pellet: A practical owl-dl reasoner. Web Semantics: Science, Services and Agents on the World Wide Web 5(2), 51–53 (2007)
8. SPARQL. W3C Recommendation, http://www.w3.org/TR/rdf-sparql-query/
9. Haarslev, V., Moller, R.: Description of the racer system and its applications. In: International Workshop on Description Logics, pp. 132–141. Stanford University, USA (2001)
10. Gennari, J., Musen, M., Fergerson, R., Grosso, W., Crubezy, M., Eriksson, H., Natalya, N., Samson, T.: The evolution of Protege: an environment for knowledge-based systems development. IJHCS 58(1), 89–123 (2003)
11. Lefort, L., Taylor, K., Ratcliffe, D.: Towards scalable ontology engineering patterns: lessons learned from an experiment based on w3c's part-whole guidelines. In: 2nd Australasian workshop on Advances in ontologies, vol. 72, pp. 31–40. Australian Computer Society, Inc., Darlinghurst (2006)
12. Tsarkov, D., Horrocks, I.: Fact++ description logic reasoner: System description. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 292–297. Springer, Heidelberg (2006)

Management of Situated Knowledge for Complex Instruments Using 3D Items Creation

Loic Merckel[1,2] and Toyoaki Nishida[1]

[1] Dept. of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501 Japan
loic@ii.ist.i.kyoto-u.ac.jp,
nishida@i.kyoto-u.ac.jp

[2] Dept. of Scientific Systems R&D, Horiba Ltd., Kyoto 601-8510 Japan

Abstract. In this paper, we present an approach to managing situated knowledge, i.e., knowledge specific to a particular location, for complex instruments. Our research is driven by real needs in the scientific instrumentation field. Although many works propose a solution for real-world annotation, those approaches might not provide the user with an effective means for representing or expressing knowledge about a complex device due to the limitation set by text note and predefined structures. With the aim of representing more elaborated situated knowledge an extended version of Spatial Knowledge Quantum is defined. The objective is to bring the possibility of enhancing the real-world with arbitrary 3D shapes. Two different interfaces for manipulating knowledge are proposed, one is based on Augmented Reality, while the second relies on Augmented Virtuality to build a virtual environment. This latter interface can be produced and distributed without extra costs, and consequently, allows knowledge dissemination at a larger scale. A preliminary version of our framework has been implemented and deployed in customers' environments.

1 Introduction and Background

In the field of complex scientific instruments, although many efforts are put into the development of intuitive and easy-to-use interfaces, some instruments remain complex to set-up, to utilize, to maintain, and so on. Although user manuals constitute one of the largest sources of knowledge about scientific instruments, the limitations are two-fold. First, even if all the relevant information is contained in the manual, without an effective index or knowledge of the structure of the mass of information, the user will likely have trouble finding it. Second, understanding the contents described in the manual might be an issue due to a lack of the vocabulary needed to describe the instrument. As a consequence a large amount of this knowledge is not communicated to the user and should be managed as a tacit form of knowledge. Indeed, several researchers consider knowledge that can be, but has not yet been, made explicit tacit knowledge [1,2]. Furthermore, a manual is generally updated only when the instrument is improved. It is thus usually a static and closed information space.

Knowledge management encompass a wide range of practices/approaches depending of the objectives and organization. Our work focuses on the knowledge required

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 459–469, 2009.

to utilize and understand a panel of complex scientific instruments, and in particular, the situated knowledge that is considered to be the knowledge specific to a particular position in the instrument space.

The recent progresses of the Augmented Reality (AR) on embedded/portable devices make such an approach realistic to be used in real-world environment with the aim of developing efficient support systems for the users. Consequently, a number of systems have been designed to solve the problem of knowledge remaining tacit, even though it can be made explicit, due to the unavailability of a chance for formalization, which is mainly due to the lack of an effective means of associating a place with specific information so that other people can find and use it. Among these initiatives we can distinguish several approaches. For instance, 3D computer generated data are superimposed onto the images so as to guide a user through performing a sequence of 3D tasks (e.g., an early prototype is described in [3]), whereas some other contributions focus on techniques to annotate the real-world (e.g., [4,5]). Although most of the existing systems assume some prior knowledge of the geometry (such as 3D model of the subject machinery), recent progresses in Simultaneous Localization And Mapping (SLAM) systems allow to annotate unknown environments [6]. Since during the instrument manufacturing process, computer-aided design (CAD) software are used to build the model of each physical/mechanical parts forming the instrument, we assume that a model is known in the scope of this work.

Fischer and Ostwald [7] define Knowledge Management (KM) as a cyclic process involving three related activities: creation, integration, and dissemination/use of knowledge. Many of the approaches based on portable AR systems related in the literature will unlikely provide an effective means for fitting into this loop. Indeed, numerous technical difficulties (e.g., computer vision, computer graphics, user interface) that must be overcome to build such an apparatus have led many researchers to focus on solving those difficulties rather than managing knowledge itself.

For instance a SLAM based approach (e.g., [6]) can allow an expert to remotely support a user by annotating the real-world without knowledge of the environment, i.e., without being constrained from a space defined by a model. As a consequence relevant knowledge is created and disseminated efficiently during the limited time-period of the procedure. However there is a lack of sustainability for the exchanged knowledge will be difficult to store so as to be reused in another environment (mainly because of the absence of a referential model). Furthermore text information or predefined shapes (such as arrows, squares and so on) may limit the possibilities of expression. A support system (e.g., [3]) that enhances the real-world using complex geometry structures requires a certain amount of preliminary work to construct and integrate those structures, making this approach falling into what Fischer and Ostwald [7] refer as "*traditional KM*" that, among other flaws, lacks of interactivity (knowledge integration is performed at "*design time, before system deployment*"). The final version of our proposed framework should implement a 3D drawing method that allows the user to efficiently draw and integrate arbitrary shapes in the scene. To reach this goal, we propose a propitious situated knowledge representation.

Moreover, such an attempt will, at one point, require industrial deployment in the customer side, and, consequently, the constraints due to production and distribution have to be considered. Mobile devices designed to augment the real-world are usually expensive for, in addition to the mobile computer, it may take advantage of expensive sensors (inclinometers, accelerometers, gyroscopes, etc.) to gain efficiency and robustness. Indeed, the scientific instruments market is very competitive and, as a consequence, cost reduction is a primary concern. This logistical limitation have to be addressed for successfully managing knowledge dissemination at a larger scale. In this regard, we propose a multi-interfaces system for manipulating knowledge. One interface is based on AR, while the second relies on Augmented Virtuality (AV) to build a virtual environment.

A major difficulty to build such interfaces is the pose computation of the subject instrument (that consists of its position and its orientation with respect to the camera coordinate system). Indeed, AR needs the pose (at run time) for registering virtual objects with the real world image. In order to build a virtual instrument, we augment the 3D model with real-world textures that are acquired from a set of images. For each of those images, a precise pose is required to extract a correct texture patch. We have proposed a solution to the pose calculation problem in previous works and we have introduced a novel object recognition algorithm [8,9] as well as a low-overhead technique for texturing 3D models [10]. In this paper we address the problem of building a situated knowledge medium based on our 3D perception suite.

The key parts of the framework have been implemented and a simplified version (that does not include the 3D drawing) has been recently deployed in customers's environments. The AV based interface has been integrated to the HORIBA EMGA-920 analyzer[1] software and exhibited with the instrument (that analyzes the quantities of oxygen, nitrogen, and hydrogen contained in steels and other metals) at the JAIMA show[2] 2008 during which it received very encouraging reactions and comments from visitors.

The remainder of this paper is organized as follow. In the next section, we introduce Mirage, the framework we use for implementing a situated knowledge management system. In section 3, we briefly present our knowledge representation. The mixed reality based interfaces for knowledge manipulation are described in section 4. We discuss some technical difficulties related to 3D drawing from 2D interface in section 5 and propose a preliminary solution. Then, in section 6, we discuss our future work, and finally, in section 7, we summarize our key points.

2 The Framework of Mirage

Mirage encompasses the knowledge management process as shown in Fig. 1. This approach follows the design perspective proposed by Fischer and Ostwald [7] in which the knowledge is created through a collaborative process, integrated at use time (and not at design time) and disseminated on demand. When a user needs to get some information about a particular component, he or she just asks for it by selecting the item of interest (via the proposed interfaces). In the eventuality of a failure to get an adequate answer for the selected position, the user is invited to ask a question that is stored in the

[1] http://www.jp.horiba.com/analy/emga-920/

[2] http://www.jaimashow.jp/english/

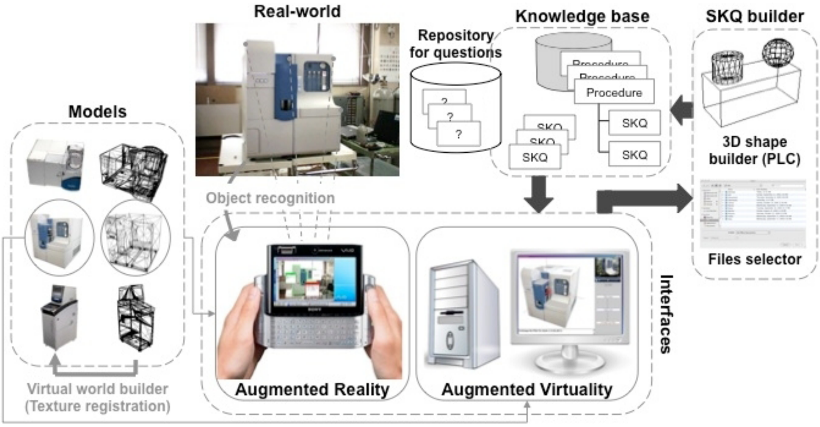


Fig. 1. Overview of Mirage

database. Subsequently, an expert is required to answer that question by creating new entries in the knowledge base.

The interface allows the user to create/edit and retrieve knowledge. We propose two different approaches to build such an interface. The first one is based on AR, while the second one is based on AV. Figure 1 presents the both interfaces. The knowledge representation is independent from those interfaces, and thus either one or the other can be used to manipulate the same knowledge base. These interfaces allow to visualize the subject instrument enhanced with adequate knowledge representation (see the next section).

The AV based interface proposes similar functionalities with the AR one. The main difference lies in the fact that the situated knowledge does not augment directly the real-world instrument, but a virtual environment (representing the instrument) is considered instead. Such an interface is easy to produce (piece of software) and to distribute (via the software dedicated to the instrument) without additional costs, whereas these statements are untrue regarding the AR approach.

A three-dimensional pointing system, allowing the user to select a 3D position from the 2D screen, is implemented by the both approaches. Concerning the AR interface, we introduced this low-cost three-dimensional pointer based on pose calculation and mesh generation of the instrument model in a previous work [5]. This approach has been adapted to the AV based interface. Since we are in a virtual environment the pose of the instrument is already precisely known, and consequently, only the mesh generation of the model needs to be performed.

The initial inputs of the overall system are only the CAD models corresponding to the instruments. Indeed, as set forth above, the knowledge base does not need to be preliminary crafted by specialists for the required knowledge will be dynamically created after deployment. Furthermore, beside the knowledge management itself, the system integrates an apparatus for efficiently texturing the 3D models so as to build the AV based interface.

The next sections will detail the knowledge representation as well as some key features of the implementation of the entire system.

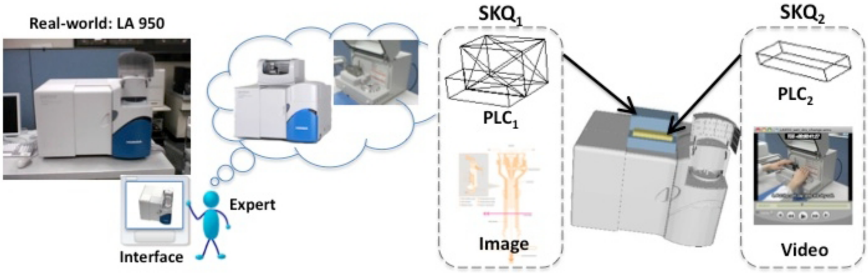


Fig. 2. The expert shares his or her knowledge about the PowderJet by creating two SKQs that encapsulate PLCs to suggest the shape and to indicate the location of the additional equipment

3 Situated Knowledge Representation

Overview. A preliminary version of the concept of Spatial Knowledge Quantum (SKQ), in which a spatial position in the environment is associated with knowledge, has been previously introduced in [5]. A SKQ consists of two parts: a three-dimensional point in the real world $p \in \mathbb{R}^3$ and the associated knowledge, which can be potentially stored in many forms: text, drawing, video and audio (it can be a combination of different items). Ordered sequences of SKQs (referred as *procedures*) can be defined and stored in the knowledge base.

Consider the scenario in which an expert would like to explain the physical principle of a particle sizer based on photon correlation spectroscopy. After opening the cover of the instrument, a basic text annotation of each components (e.g., monochromator, lenses, photomultiplier tube, cell and so on) may limit the explanation. Instead, the expert may want to draw the optical path and incorporate new imaginary 3D elements (e.g., lenses, cells) to express and communicate his or her knowledge.

Let us consider another situation in which the expert want to provide information about optional equipments and accessories. For instance, the HORIBA Partica LA-950 Particle Size Analyzer[3] comes with several options including the PowderJet Dry Feeder[4]. Suppose that the expert would like to explain the advantages and utilization of this option to users that already own the basic instrument. Text explanations, pictures or video files might be unsatisfactory, especially if a user is interested by asking a question about a particular position. The expert may prefer draw an interactive 3D sketch of the additional device integrated with the instrument (Fig. 2).

Extended spatial knowledge quantum. In this paper we extend the concept of SKQ to include arbitrary 3D shapes. The point p introduced in the previous definition can be replaced by a three-dimensional piecewise linear complexes. Piecewise Linear Complexes (PLC) were introduced by Miller et al. [11]. A simpler description, limited to three dimensions, has also been given [12,13]. A PLC is a general boundary description for three-dimensional objects and can be simply defined as a set of vertices, segments, and facets. Each facet is a polygonal region; it may have any number of sides and may

[3] http://www.jobinyvon.com/LA-950V2

[4] http://www.jobinyvon.com/PowderJet

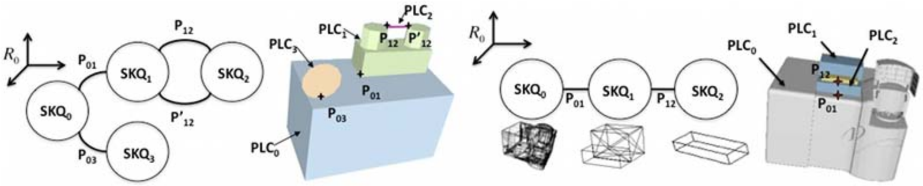


Fig. 3. Example of global knowledge structure, graph of SQKs with a 3D representation. (left) Arbitrary example. (Right) Example corresponding to Fig. 2. PLC_0 is the model of the subject instrument.

be non-convex, possibly with holes, segments, and vertices. PLCs have restrictions. For PLC X, the elements of X must be closed under intersection. For example, two segments can intersect only at a common vertex that is also in X. Two facets of X may intersect only at a shared segment or vertex or in a union of shared segments and vertices. Another restriction is that the set of points used to define a facet must be coplanar.

Our choice to restrict the 3D structure to a PLC lie in the fact that such a structure exhibit adequate properties to perform a powerful mesh generation algorithm (Refinement of Constrained Delaunay Triangulation [14]). As a result, the three-dimensional pointing system introduced in [5] can be directly use to select a precise position on a given SKQ.

Regarding the new SKQ definition, the model defining the instruments can be interpreted as a particular SKQ, which is hidden in the rendered scene and serve as a root structure that provides the reference coordinate system in which all other SKQs will be defined. We can associate new SKQs to any existing SKQ.

A graph can be used to represent the global structure of the SKQs (as illustrated in Fig. 3). A given SKQ_i consists of a PLC_i and an associated set of multimedia files (e.g., text and video) that may be empty (in such case the PLC_i should have a meaning from his shape). In order to associate two SKQs, SKQ_i and SKQ_j, we link them at a common element $P_{ij} \in PLC_i \cap PLC_j$.

This extended definition of SKQ is compatible with the previous one presented in [5]. Indeed, A previous SKQ can be interpreted as a new SKQ_i such that the associated PLC_i is reduced to a point $p \in \mathbb{R}^3$.

4 Mixed Reality Based Interfaces

Augmented Reality. The interface based on AR technology has already been partially described in a previous work [5]. The augmentation is performed on image snapshots rather than live video. The system is composed of a tablet PC equipped with a video camera and held by the user. While this anterior version relied on markers attached to the instrument for the purpose of recognizing the unit (pose calculation), the current system has been improved by implementing our novel 3D model based object recognition method [9]. Figure 4 shows few screenshots in which we can see the HORIBA EMGA-920 analyzer augmented with SKQs. The recognition method is using line features and is based on a precise 3-DOF orientation sensor attached to the camera. It consists of

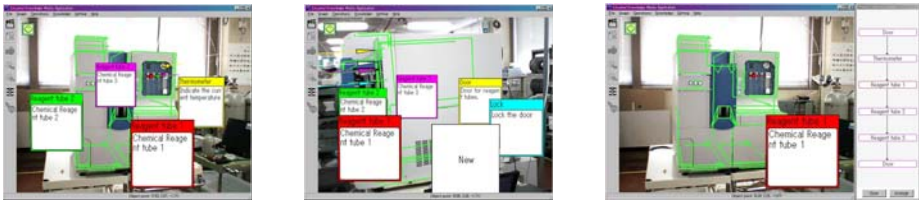


Fig. 4. Example of the current AR interface. The EMGA unit is augmented with SKQs. The green model superimposed on the image is useful for the user to determine if the object recognition result is correct. Each SKQ (colored box) is only partially shown, it has to be selected so as to be fully displayed. The right image shows a procedure in which each step represents a particular SKQ.

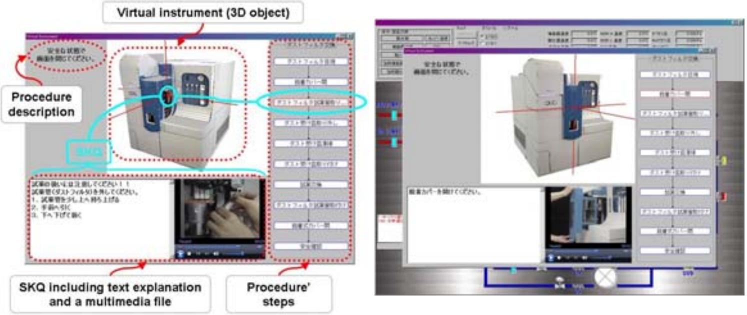

Virtual instrument (3D object)
Procedure description
SKQ
SKQ including text explanation and a multimedia file
Procedure' steps

Fig. 5. Example of the current AV interface. **Left:** Description of the interface. **Right:** Integration of the interface inside the EMGA software.

three steps: generation of pose hypotheses, evaluation of each pose to generate a ranked list, and finally, refinement of the best candidate. Although a set of experiments, conducted in [9], have demonstrated the robustness of this approach, the first ranked pose is not always a correct result (Fig. 6). Therefore, current implementation of the system gives to the user the possibility to select the best pose among the list of top ranked pose hypothesis. To do so, a superimposition of the re-projected model is perform for each candidate pose. Concerning the response time, the median value is about 3 *s*, which is acceptable.

Augmented Virtuality. Figure 5 shows an implementation of the AV based interface. That interface has been incorporated inside the HORIBA EMGA-920 software, which has been recently shipped to customers.

This virtual environment must represent the real-world instrument so as to be as effortless as possible for the user to interpret. For the purpose of reaching that target, we introduced a method for generating model texture from real-world images [10]. It is based on the object recognition method mentioned above. Figure 7 shows some models we have textured using our approach and Table 1 summarizes the results of the texture reconstructions for each model. The reconstructed objects have a texture that covers between 70.5 % to 77.6 % of the total surface area. Considering that the bottoms of

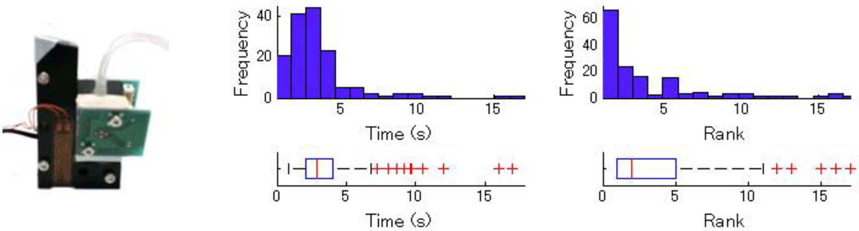


Fig. 6. CRP module (Left). Results for 150 recognitions of the CRP module performed in divers conditions such as partial occlusion and cluttered environment (Right). Below the distribution the corresponding box-and-whisker diagram is shown.

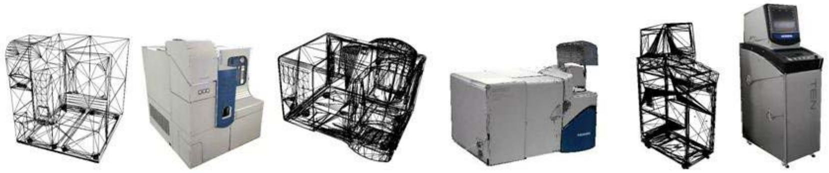


Fig. 7. Example of models we have reconstructed using our method: EMGA, LA-950, Tensec

Table 1. Texture reconstruction results for the three objects

	LA-950	EMGA	Tensec
Number of triangles	21 667	2 962	29 333
Total surface area (cm^2)	27 305	33 290	54 457
Total surface textured	74.8 %	77.6 %	70.5 %
Approx. time (*min.*)	⩽ 180	⩽ 60	⩽ 120

the objects are not textured (the position of the objects remains fixed), these results are adequate for building a virtual instrument.

In the domain of scientific instruments, the instrument may incorporate some items that are not shown on the model (e.g., cables, plugs, etc.). These items will then appear on the textured model, which is actually better to construct a realistic virtual instrument (indeed, such items will mostly be present on each operating instrument).

5 Building Spatial Knowledge Quantum

The difficulty to build SKQs resides in the generation and the integration of the associated three-dimensional PLCs from a 2D interface with low overhead. Indeed, although numerous 3D modeling softwares allow to produce precise and complex shapes of various kinds and are used for a wide range of applications, the user interfaces remain generally complex enough to prevent a non-experienced user from quickly grasping even the basics functionalities. The fact that those interfaces are based on the WIMP paradigm has been criticized and considered as unnatural for 3D tasks [15].

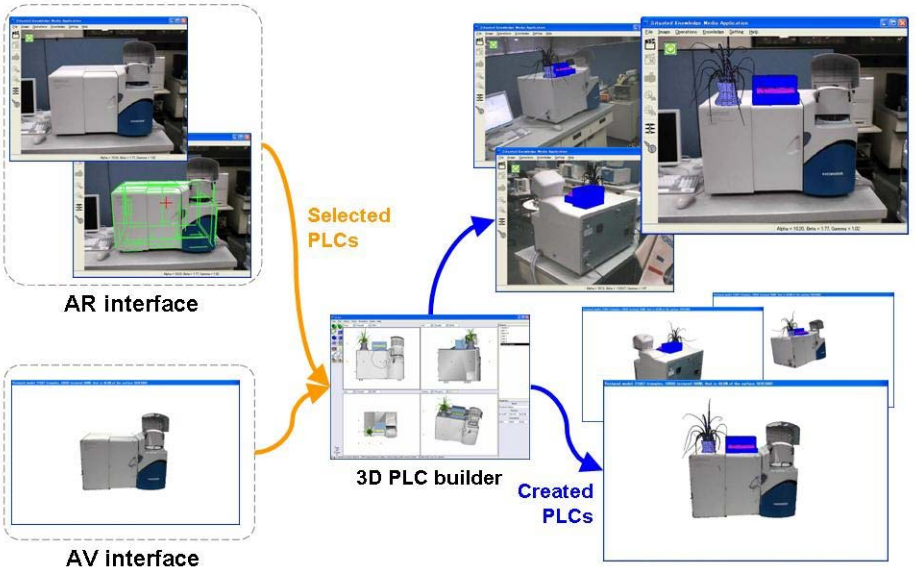


Fig. 8. Workflow of the current implementation for PLCs creation (snapshots are from our current interfaces). The PLC builder used is *Art of Illusion*.

Alternatively many works have proposed new types of interfaces, such as gestural and suggestive (e.g., [16,17,18,19]), for the purpose of drawing rough 3D objects in an *effortless* manner.

Drawing. The target is not to manufacture the object, but to help the user to express his or her idea so as to share it with others. In this context, the precision is not a main concern. Complex shapes can be approximated by a combination of simple primitives. From our experience, WIMP based 3D modeling approach can be efficient to draw and to assemble primitives such as cube, sphere, cylinder and so on. Currently, we use *Art of Illusion*[5] that fits our needs. Indeed, although it does not provide as much functionalities as commercial alternatives, it presents the advantage to be very simple to operate (without any consultation of the user manual, only few minutes were enough to understand and create the two PLCs shown in Fig. 2) and appears to be enough for drawing fairly complex PLCs. However, to manipulate a 3D component through a multi-panels view (e.g., top, front, sides, that is a typical configuration for WIMP based 3D modeling software) can be conceptually difficult for not well prepared users. Instead, gestural interfaces might be of significant interest (see the future works section).

Integration. Here we discuss the placement of the PLC in the current scene. A new PLC_j is defined relatively to another PLC_i (possibly several PLCs). The 3D pointing system allows to select PLC_i and to determine the point of connection $P_{ij} \in PLC_i$ from the 2D interface (AR or AV). All PLCs are defined with regard to the reference coordinate system $\mathcal{R}_0 = (O, \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w})$ used to build the instrument model. This root model

[5] http://www.artofillusion.org/

(PLC_0) as well as the selected PLC_i (if different from PLC_0) are exported (currently STL format) and imported in the CAD software with which the new PLC_j is created, relatively to PLC_0 and PLC_i, before being exported and re-imported in our system. Figure 8 gives an illustration in which the two PLCs shown in Fig. 2 as well as a new PLC (a model of a plant) are created and shown via our both interfaces (AR and AV).

6 Future Works

The current prototype implements a preliminary functional version of the both AR and AV interfaces, however, the creation of new SKQs requires an external CAD software when the associated PLCs are not reduced to a single point (i.e., previous definition presented in [5]). In particular, this approach is not well designed for the AR based interface, that should be run on a tablet PC using a stylus. Moreover, such an approach may prevent the user from taking advantage of 3D items, leaving this feature to specialists only. Future work includes the elaboration of a method to draw and to integrate imaginary 3D objects in the scene directly from the proposed AR or AV interfaces. Possible direction is to implement an intuitive 3D modeling interface such as the one proposed by SketchUp[6], which can be interpreted as a combination of a WIMP and a gestural interface. A fully-gestural interface (e.g., [18,17])) might also be envisaged. Actually, the final version should let the user select between several choices.

7 Conclusion

We have presented our approach to implementing a situated knowledge management system in the limited area of complex scientific instruments. A new representation of situated knowledge has been introduced and two different interfaces, one based on AR and the other one based on AV, for knowledge manipulations have been described. A first version of the AV based interface has been recently integrated in a commercial software, and is being used by customers.

References

1. Gourlay, S.: Tacit knowledge, tacit knowing, or behaving? In: 3rd European Organizational Knowledge, Learning, and Capabilities, Athens, Greece (2002)
2. Hedesstrom, T., Whitley, E.A.: What is meant by tacit knowledge? towards a better understanding of the shape of actions. In: The 8th European Conference on Information Systems (ECIS) (2000)
3. Feiner, S., Macintyre, B., Seligmann, D.: Knowledge-based augmented reality. Commun. ACM 36(7), 53–62 (1993)
4. Rose, E., Breen, D., Ahlers, K.H., Crampton, C., Tuceryan, M., Whitaker, R., Greer, D.: Annotating real-world objects using augmented reality. In: Proceedings of Computer Graphics International, Leeds, UK, pp. 357–370 (1995)

[6] http://sketchup.google.com/

5. Merckel, L., Nishida, T.: Enabling situated knowledge management for complex instruments by real-time reconstruction of surface coordinate system on a mobile device. AI & Society (2008) (accepted for publication)
6. Reitmayr, G., Eade, E., Drummond, T.: Semi-automatic annotations in unknown environments. In: Proc. ISMAR 2007, Nara, Japan, November 13–16, pp. 67–70 (2007)
7. Fischer, G., Ostwald, J.: Knowledge management: Problems, promises, realities, and challenges. IEEE Intelligent Systems 16(1), 60–72 (2001)
8. Merckel, L., Nishida, T.: Evaluation of a method to solve the perspective-two-point problem using a three-axis orientation sensor. In: IEEE the 8th International Conference on Information Technology (CIT), Sydney, Australia, pp. 862–867. IEEE Computer Society, Los Alamitos (2008)
9. Merckel, L., Nishida, T.: Accurate object recognition using orientation sensor with refinement on the lie group of spatial rigid motions. IEICE Transactions on Information and Systems E91-D(8), 2179–2188 (2008)
10. Merckel, L., Nishida, T.: Multi-interfaces approach to situated knowledge management for complex instruments: First step toward industrial deployment. In: The 7th International Workshop on Social Intelligence Design, San Juan, Puerto Rico (2008) (to be presented)
11. Miller, G., Talmor, D., Teng, S., Walkington, N., Wang, H.: Control volume meshes using sphere packing: Generation, refinement and coarsening. In: Fifth International Meshing Roundtable, Pittsburgh, Pennsylvania, pp. 47–61 (1996)
12. Shewchuk, J.R.: Tetrahedral mesh generation by delaunay refinement. In: Symposium on Computational Geometry, pp. 86–95 (1998)
13. Si, H., Gaertner, K.: Meshing piecewise linear complexes by constrained delaunay tetrahedralizations. In: Proceedings of the 14th International Meshing Roundtable, pp. 147–163 (2005)
14. Si, H.: On refinement of constrained delaunay tetrahedralizations. In: Proceedings of the 15th International Meshing Roundtable (2006)
15. van Dam, A.: Post-wimp user interfaces. Communications of the ACM 40(2), 63–67 (1997)
16. Zeleznik, R.C., Herndon, K.P., Hughes, J.F.: Sketch: an interface for sketching 3d scenes. In: SIGGRAPH 1996: Proceedings of the 23rd annual conference on Computer Graphics and Interactive Techniques, pp. 163–170. ACM Press, New York (1996)
17. Igarashi, T., Matsuoka, S., Tanaka, H.: Teddy: A sketching interface for 3D freeform design. In: ACM SIGGRAPH 1999, pp. 409–416 (1999)
18. Igarashi, T., Hughes, J.F.: A suggestive interface for 3D drawing. In: 14th Annual Symposium on User Interface Software and Technology, Orlando, Florida, pp. 173–181 (2001)
19. Masry, M., Kang, D.J., Lipson, H.: A freehand sketching interface for progressive construction of 3D objects. Computers & Graphics 29(4), 563–575 (2005)

Two Applications of Lexical Information to Computer-Assisted Item Authoring for Elementary Chinese

Chao-Lin Liu, Kan-Wen Tien, Yi-Hsuan Chuang, Chih-Bin Huang, and Juei-Yu Weng

Department of Computer Science, National Chengchi University, Taiwan
{chaolin,g9627,s9436,g9614,s9403}@cs.nccu.edu.tw

Abstract.[1] Testing is a popular way to assess one's competence in a language. The assessment can be conducted by the students for self evaluation or by the teachers in achievement tests. We present two applications of lexical information for assisting the task of test item authoring in this paper. Applying information implicitly contained in a machine readable lexicon, our system offers semantically and lexically similar words to help teachers prepare test items for cloze tests. Employing information about structures and pronunciations of Chinese characters, our system provides characters that are similar in either formation or pronunciation for the task of word correction. Experimental results indicate that our system furnishes quality recommendations for the preparation of test items, in addition to expediting the process.

Keywords: Computer assisted test-item authoring, Chinese synonymy, Chinese-character formation, natural language processing.

1 Introduction

The history of applying computing technologies to assisting language learning and teaching can be dated back as least 40 years ago, when the Programmed Logic for Automatic Teaching Operations, which is referred as PLATO usually, was initiated in 1960 [6; p. 70]. The computing powers of modern computers and the accessibility to information supported by the Internet offer a very good environment for language learning that has never been seen before.

The techniques for natural language processing (NLP) [11] are useful for designing systems for information retrieval, knowledge management, and language learning, teaching, and testing. In recent years, the applications of NLP techniques have received attention of researchers in the Computer Assisted Language Instruction Consortium (often referred as CALICO, http://calico.org/, instituted in 1983) and the

[1] Due to the subject matter of this paper, we must show Chinese characters in the text. Whenever appropriate, we provide the Chinese characters with their Romanized forms and their translations in English. We use traditional Chinese and Hanyu Pinyin, and use Arabic digits to denote the tones in Mandarin. (http://en.wikipedia.org/wiki/Pinyin) For readers who do not read Chinese, please treat those individual characters as isolated pictures or just symbols.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 470–480, 2009.

researchers in the computational linguistics, e.g., in United States of America [13] and in Europe [5]. Heift and Schulze [6] report that there are over 100 documented projects that employed NLP techniques for assisting language learning.

The applications of computing technologies to the learning of Chinese language can also be traced back as far as 40 years ago, when researchers applied computers to collate and present Chinese text for educational purposes [14]. The superior computing powers of modern computers offer researchers and practitioners to invent more complicated tools for language learning. As a result, such computer-assisted language learning applications are no longer limited to academic laboratories, and have expanded their existence into real-world classrooms [1, 16].

In this paper, we focus on how computers may help teachers assess students' competence in Chinese, and introduce two new applications for assisting teachers to prepare test items for elementary Chinese. Students' achievements in cloze tests provide a good clue to whether they learned the true meanings of the words, and the ability to identify and correct a wrong word in the so-called word-correction tests is directly related to students' ability in writing and reading. Our system offers semantically or lexically similar words for preparing the cloze items, and provides characters that are visually or phonetically similar to the key characters for the word-correction items. Experimental results indicate that the confusing characters that our system recommended were competitive in quality, even when compared with those offered by native speakers of Chinese.

We explain how to identify semantically and lexically similar Chinese words in Section 2, elaborate how to find structurally and phonetically similar Chinese characters in Section 3, and report an empirical evaluation of our system in Section 4. Finally, we make concluding remarks in Section 5.

2 Semantically and Lexically Similar Words for Cloze Tests

A cloze test is a multiple-choice test, in which one and only one of the candidate words is correct. The examinee has to find the correct answer that fits the blank position in the sentence. A typical item looks like the following. (A translation of the sentence used in this test item is "The governor officially ___ the Chinese teachers' association yesterday, and discussed with the chairperson about the education problems for the Chinese language in California." The four choices, including the answer, are different ways to say "visit" or "meet" in Chinese.)

> 州長於昨日正式____中文教師學會，與會長深入討論加州的中文教育問題。(a) 見面 (b) 走訪 (c) 拜訪 (d) 訪視

Cloze tests are quite common in English tests, such as GRE and TOEFL. Applying techniques for word sense disambiguation, Liu et al. [10] reported a working system that can help teachers prepare items for cloze tests for English. Our system offers a similar service for Chinese cloze tests.

To create a cloze test item, a teacher determines the word that will be the answer to the test item, and our system will search in a corpus for the sentences that contain the answer and present these sentences to the teacher. The teacher will choose one of these sentences for the test item, and our system will replace the answer with a blank area in the sample sentence (the resulting sentence is usually called *stem* in computer assisted item generation), and show an interface for more authoring tasks.

A cloze item needs to include distracters, in addition to the stem and the correct answer in the choices. To assist the teachers prepare the distracters, we present two types of candidate word lists to the teachers. The first type of list includes the words that are semantically similar to the answer to the cloze item, and the second type of list contains the words that are lexically similar.

We have two sources to obtain semantically similar words. The easier way is to rely on a Web-based service offered by the Institute of Linguistics at the Academia Sinica to find Chinese words of similar meanings [3], and present these words to the teachers as candidates for the distracters. We have also built our own synonym finder with HowNet (http://www.keenage.com). HowNet is bilingual machine readable lexicon for English and Chinese. HowNet employs a set of basic semantic units to explain Chinese words. Overlapping basic semantic units of two Chinese words indicate that these words share a portion of their meanings. Hence, we can build a synonym finder based on this observation, and offer semantically related words to the teachers when they need candidate words for the distracters of the cloze items. In addition, words that share more semantic units are more related than those that share fewer units. Hence, there is a simple way to prioritize multiple candidate words.

When assisting the authoring of cloze items, we can obtain lists of Chinese words that are semantically similar to the answer to the cloze item with the aforementioned methods. For instance, "造訪" (zao(4) fang(3)), "拜會" (bai(4) hui(4)), and "走訪" (zou(3) fang(3)) carry a similar meaning with "拜訪" (bai(4) fang(3)). The teachers can either choose or avoid those semantically similar, yet possibly contextually inappropriate in ordinary usage, words for the test items.

It is the practice for teachers in Taiwan to use lexically similar words as distracters. For this reason, our system presents words that contain the same characters with the answer as possible distracters. For instance, both "喝酒" (he(1) jiu(3)) and "奉茶" (feng(4) cha(2)) can serve as a distracter for "喝茶" (he(1) cha(2)) because they share one character at exactly the same position in the words. We employ HowNet to find candidate words of this category.

3 Visually and Phonetically Similar Words for Word Correction

In this section, we explain how our system helps teachers prepare test item for "word correction." In this type of tests, a teacher intentionally replaces a Chinese character with an incorrect character, and asks students to identify and correct this incorrect character. A sample test item for word correction follows. (A translation of this Chinese string: The wide varieties of the exhibits in the flower market dazzle the visitors.)

> 花市中各種展品讓人眼花繚亂 ("繚" is incorrect, and should be replaced with "撩")

Such an incorrect character is typically similar to the correct character either visually or phonetically. Since it is usually easy to find information about how a Chinese character is uttered, given a lexicon, we turn our attention to visually similar characters. Visually similar characters are important for learning Chinese. They are also important in the psychological studies on how people read Chinese [12, 15]. We present some similar Chinese characters in the first subsection, illustrate how we encode Chinese characters in the second subsection, elaborate how we improve the encoding method to facilitate the identification of similar characters in the third subsection, and discuss the weakness of our current approach in the last subsection.

3.1 Examples of Visually Similar Chinese Characters

士土工干千 戊戌成 田由甲申
母毋 勿匆 人入 未末 采釆 凹凸

Fig. 1. Some similar Chinese characters

頸勁 構溝 陪倍 硯現 裸棵 搞篙
列刑 盆盎盂盅 因困囚 間閒閃開

Fig. 2. Some similar Chinese characters that have different pronunciations

形刑型 踵種腫 購構搆 紀記計
園圓員 脛逕徑痙勁

Fig. 3. Homophones with a shared component

We show three categories of similar Chinese characters in Figures 1, 2, and 3. Groups of similar characters are separated by spaces in these figures. In Figure 1, characters in each group differ at the stroke level. Similar characters in every group in the first row in Figure 2 share a common component, but the shared component is not the radical of these characters. Similar characters in every group in the second row in Figure 2 share a common component, which is the radical of these characters. Similar characters in every group in Figure 2 have different pronunciations. We show six groups of homophones that also share a common component in Figure 3. Characters that are similar in both pronunciations and internal structures are most confusing to new learners.

It is not difficult to list all of those characters that have the same or similar pronunciations, e.g., "試" and "市", if we have a machine readable lexicon that provides information about pronunciations of characters and when we ignore special patterns for tone sandhi in Chinese [2].

In contrast, it is relatively difficult to find characters that are written in similar ways, e.g., "構" with "購", with an efficient manner. It is intriguing to resort to image processing methods to find such structurally similar words, but the computational costs can be very high, considering that there can be tens of thousands of Chinese characters. There are more than 22000 different characters in Chinese [7], so directly computing the similarity between images of these characters demands a lot of computation. There can be more than 242 million combinations of character pairs. The Ministry of Education in Taiwan suggests that about 5000 characters are needed for everyday communication. In this case, there are about 12.5 million pairs.

The quantity of combinations is just one of the bottlenecks. We may have to shift the positions of the characters "appropriately" to find the common component of a character pair. The appropriateness for shifting characters is not easy to define, making the image-based method less directly useful; for instance, the common component of the characters in the rightmost group in the second row in Figure 3 appears in different places in the characters.

Lexicographers employ radicals of Chinese characters to organize Chinese characters into sections in dictionaries. Hence, the information should be useful. The groups in the second row in Figure 3 show some examples. The shared components in these groups are radicals of the characters, so we can find the characters of the same group in the same section in a Chinese dictionary. However, information about radicals as they are defined by the lexicographers is not sufficient. The groups of characters shown in the first row in Figure 3 have shared components. Nevertheless, the shared components are not considered as radicals, so the characters, e.g., "頸"and "勁", are listed in different sections in the dictionary.

3.2 Encoding the Chinese Characters with the Cangjie Codes

The Cangjie method is one of the most popular methods for people to enter Chinese into computers. The designer of the Cangjie method, Mr. Chu, selected a set of 24 basic elements in Chinese characters, and proposed a set of rules to decompose Chinese characters into these elements [4]. Hence, it is possible to define the similarity between two Chinese characters based on the similarity between their Cangjie codes.

Table 1 has three sections, each showing the Cangjie codes for some characters in Figures 1, 2, and 3. Every Chinese character is decomposed into an ordered sequence of elements. (We will find that a subsequence of these elements comes from a major component of a character, shortly.) Evidently, computing the number of shared elements provides a viable way to determine "visual similarity" for characters that appeared in Figures 2 and 3. For instance, we can tell that "搞" and "篙" are similar because their Cangjie codes share "卜口月", which in fact represent "高".

Table 1. Cangjie codes for some characters

	Cangjie Codes		Cangjie Codes
士	十一	土	土
工	一中一	干	一十
勿	心竹竹	匆	竹田心
未	十木	末	木十
頸	一一一月金	勁	一一大尸
硯	一口月山山	現	一土月山山
搞	手卜口月	篙	竹卜口月
列	一弓中弓	刑	一廿中弓
因	田大	困	田木
間	日弓日	閒	日弓月
踵	口一竹十土	種	竹木竹十土
腫	月竹十土	紀	女火尸山
購	月金廿廿月	構	木廿廿月
記	卜口尸山	計	卜口十
圓	田口月金	員	口月山金
脛	月一女一	逕	卜一女一
徑	竹人一女一	痙	大一女一

Unfortunately, the Cangjie codes do not appear to be as helpful for identifying the similarities between characters that differ subtly at the stroke level, e.g., "士土工干" and others listed in Figure 1. There are special rules for decomposing these relatively basic characters in the Cangjie method, and these special encodings make the resulting codes less useful for our tasks.

The Cangjie codes for characters that contain multiple components were intentionally simplified to allow users to input Chinese characters more efficiently. The average number of key strokes needed to enter a character is a critical factor in designing input methods for Chinese. The longest Cangjie code among all Chinese characters contains five elements. As shown in Table 1, the component "巠" is represented by "一女一" in the Cangjie codes for "脛" and "徑", but is represented only by "一一" in the codes for "頸" and "勁". The simplification makes it relatively harder to identify visually similar characters by comparing the actual Cangjie codes.

3.3 Engineering the Cangjie Codes for Practical Applications

Though useful for the design of an input method, the simplification of Cangjie codes causes difficulties when we use the codes to find similar characters. Hence, we choose to use the complete codes for the components in our database. For instance the complete codes for "巠", "脛", "徑", "頸", and "勁" are, respectively, "一女女一", "月一女女一", "竹人一女女一", "一女女一一月山金", and "一女女一大尸".

The information about the structures of the Chinese characters [7, 9] can be instrumental as well. Consider the examples in Figure 3. Some characters can be decomposed vertically; e.g., "盅" can be split into two smaller components, i.e., "中" and "皿". Some characters can be decomposed horizontally; e.g., "現" is consisted of "王" and "見". Some have enclosing components; e.g., "人" is enclosed in "口" in "囚". Hence, we can consider the locations of the components as well as the number of shared components in determining the similarity between characters.

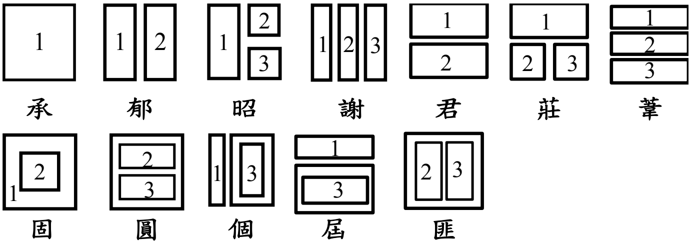


Fig. 4. Layouts of Chinese characters (used in Cangjie)

Figure 4 illustrates the layouts of the components in Chinese characters that were adopted by the Cangjie method [9]. A sample character is placed below each of these layouts. A box in a layout indicates a component, and there can be at most three components in a character. We use digits to indicate the ordering the components. Due to space limits in the figure, we do not show all digits for the components.

After recovering the simplified Cangjie code for a character, we can associate the character with a tag that indicates the overall layout of its components, and separate the code sequence of the character according to the layout of its components. Hence, the information about a character includes the tag for its layout and between one to three sequences of code elements. The layouts are numbered from left to right and from top to bottom in Figure 4. Table 2 shows the annotated and expanded codes of the sample

Table 2. Some annotated and expanded codes

	Layout	Comp. 1	Comp. 2	Comp. 3
承	1	弓弓手人		
郁	2	大月	弓中	
昭	3	日	尸竹	口
謝	4	卜一一口	竹難竹	木戈
君	5	尸大	口	
莊	6	廿	女中一	土
葦	7	廿	木一	手
因	8	田	大	
國	9	田	戈	口一
頸	2	一女女一	一月山金	
徑	2	竹人	一女女一	
員	5	口	月山金	
圓	9	田	口	月山金
相	2	木	月山	
想	5	木月山	心	
箱	6	竹	木	月山

characters in Figure 4 and the codes for some characters that we will discuss. Elements that do not belong to the original Cangjie codes of the characters are shown in a bounding box.

Recovering the elements that were dropped out by the Cangjie method and organizing the sub-sequences of elements into components facilitate the identification of similar characters. It is now easier to find that the character (頸) that is represented by "一女女一" and "一月山金" looks similar to the character (徑) that is represented by "竹人" and "一女女一" in our database than using their original Cangjie codes in Table 1. Checking the codes for "員" and "圓" in Table 1 and Table 2 will offer an additional support for our design decisions.

Computing the similarity between characters using a database of such strengthened Cangjie code is very efficient. In the worst case, we need to compare nine pairs of code sequences for two characters that both have three components. Since we are just doing simple string comparisons, computing the similarity between characters is simple. It takes less than one second to find visually similar characters from a list of 5000 characters on a Pentium IV 2.8GHz CPU with 2G RAM. Moreover, we can offer a search service that allows psycholinguistics researchers to look for characters that contain specific components that locate at particular places within the characters.

3.4 Drawbacks of Using the Cangjie Codes

Using the Cangjie codes as the basis for comparing the similarity between characters introduces some potential problems.

It appears that the Cangjie codes for some characters, particular those simple ones, were not assigned without ambiguous principles. Relying on the Cangjie codes to compute the similarity between such characters can be difficult. For instance, "分" uses the fifth layout, but "兌" uses the first layout in Figure 4. The first section in Table 1 shows the Cangjie codes for some character pairs that are difficult to compare. It appears that we need to mark the similarity among such special characters manually, perhaps with the interactive assistance of the methods proposed in this paper.

Except for the characters that use the first layout, the Cangjie method splits characters into two or three components, and considers one of these components more important than the others. For the characters that use layouts 2, 3, 4, 5, 6, 7, 10 or 11, the component that locates at the left-most side or at the top of the layout is the most important. For the characters that use layouts 8, 9, or 12, the bounding box is the most important. In Figure 4, components that are marked "1" are the most important ones.

Due to this design principle of the Cangjie codes, there can be at most one component at the left hand side and at most one component at the top in the layouts. The last three entries in Table 2 provide an example for these constraints. As a standalone character, "相" uses the second layout. Like the standalone "相", the "相" in "箱" was divided into two parts. However, in "想", "相" is treated as an individual component because it is on top of "想". Similar problems may occur elsewhere, e.g., "森焚" and "恩因". There are also some exceptional cases; e.g., "品" uses the sixth layout, but " 闆" uses the fifth layout.

Just like that we can choose not to simplify codes for components as we discussed in Section 3.3, we can choose not to follow this Cangjie's rule about layouts of Chinese characters. This is a feasible design choice, and we plan to implement the idea.

4 Evaluation with Real Test Items

We put into field tests the methods reported in Section 3 to show their practicability. Table 3 shows 20 Chinese words that can be included in test items for the task of word correction. These words were arbitrarily chosen from a book that was written for learning correct Chinese [17]. The underlined character in each of these 20 words is the character that is often written incorrectly. For each of these 20 words, the book provides the incorrect character that is most commonly used to replace the correct character. To make the explanation succinct in this section, we refer to these underlined characters as *target characters*. Given the words in Table 3, 21 native speakers of Chinese were asked to write down one character for each of these target characters.

We employed the method reported in Section 3 to find visually similar characters from 5000 Chinese characters. We selected phonetically similar characters for the target characters based on the list of confusing Chinese sounds that is provided by a psycholinguist of the Academia Sinica [8].

4.1 Prioritizing the Candidate Characters with Real-World Information

In the experiments, we had the flexibility to allow our system to recommend a particular number of incorrect characters for the target characters. We must have a way to prioritize the candidate characters so that we can recommend a particular number of candidate characters that may meet the teachers' expectation. More specifically, if the teachers want our system to recommend no more than 10 candidate characters and if our system has more than 10 candidate characters, how does our system choose from all the candidate characters?

This is an interesting question that requires the expertise in psycholinguistics. Facing this problem, a typical psycholinguist will probably refer us to the literature on how human read Chinese text. Indeed, computer scientists may try to do some experiments and apply machine learning techniques to learn the concept of "degree of confusion" from the collected data.

We take advantage of Google for this task. Take the number 3 item in Table 3 for example. Assume that we want to prioritize "僚", "瞭", and "繚" for "撩". We can search

Table 3. Chinese words used in the evaluation

item #	word	item #	word	item #	word	item #	word
1	一剎那	2	一炷香	3	眼花撩亂	4	相形見絀
5	作踐	6	剛愎自用	7	可見一斑	8	和藹可親
9	彗星	10	萎靡不振	11	穠纖合度	12	待價而沽
13	獎券	14	意興闌珊	15	罄竹難書	16	搔首弄姿
17	根深柢固	18	椿萱並茂	19	煩躁	20	璀璨

“眼花僚亂”, “眼花瞭亂”, and “眼花繚亂” in Google, and see the number of pages that used these words. Note that we must use the double quotations in our queries so that Google searches the words verbatim. On the date of this writing, Google reports that there are, respectively, 2650, 37100, and 118000 pages for these three queries. Hence, when asked for just one recommendation, our system will return “繚”; and, when asked for two recommendations, our system will return “繚” and “瞭”.

4.2 Experimental Results: Competing with Native Speakers

Evidence showed that the native speakers who participated in our experiments did not agree with each other very well. Since every human subject returned a list of 20 characters for the words in Table 3, we collected a total of 21 lists of 20 characters. We used one of these lists as the “correct” answer and checked how the remaining 20 lists agreed with the correct answer. We repeated this process 21 times, and recorded the number of agreed character pairs. Because there were 21 human subjects, there were (21×20÷2)=210 pairs of human subjects. The agreement between any human subject pair ranged between 0 and 20.

Figure 5 shows how these human subjects agreed with each other. The horizontal axis shows the number of characters that appeared in the lists of a pair a human subject, and the vertical axis shows the number of human subject pairs that agreed on a particular number, indicated on the horizontal axis, of character pairs. On average, a human subject agreed with other human subjects only on 8.905 characters.

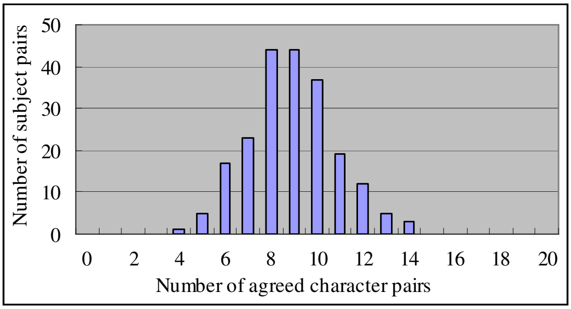


Fig. 5. Human subjects did not agree very well

How well did our system perform? We allowed our system to recommend only one character for the words in Table 3, and compared the recommended characters with the lists provided by the book [17] that we relied on. Out of the 20 characters, our system provided 13 perfect answers. In contrast, answers provided by the best performing human subjects contained 14 perfect answers, but the average of all 21 human subjects achieved only 10.66 characters.

We also evaluated the performance of our system with the precision and recall measures that are popular in the literature on information retrieval [11]. Again, we used the answers provided in the book [17] as the perfect answers. Asked to provided five candidate characters for the words in Table 3, our system achieved 0.19 and 0.875, respectively, in precision and recall rates. Notice that a precision of 0.19 is very good in this experiment, because our system was forced to offer five candidate characters while there was only one perfect answer. A precision of 0.19 suggests that our system almost caught the perfect answer with just five candidate characters.

5 Concluding Remarks

We report the applications of phonetic, lexical, and semantic information to the design of a computer-assisted item authoring environment. Our experience indicates that the environment can improve the efficiency for item authoring. Experimental results further show that the resulting system is useful for preparing quality test-items. Nevertheless, we must note that the quality of the compiled test items depends highly on the experts who actually prepare the test items. The capability of finding visually similar characters is also very useful for conducting psycholinguistic studies, and we have begun a joint research project in this direction.

It is important to study the effects of employing the incorrect characters in real-world applications that involve human subjects to investigate their reactions. We have taken steps toward this research, and will report the findings in an extended report.

Acknowledgments. We thank anonymous reviewers for their invaluable comments, and we will respond to the comments in an extended report. This research project was funded in part by the grants NSC-95-2221-E-004-013-MY2 and NSC-97-2221-E-004-07-MY2 from the National Science Council of Taiwan.

References

1. Bourgerie, D.S.: Computer assisted language learning for Chinese: A survey and annotated bibliography. J. of the Chinese Language Teachers Association 38(2), 17–48 (2003)
2. Chen, M.Y.: Tone Sandhi: Patterns across Chinese Dialects. Cambridge Studies in Linguistics, vol. 92. Cambridge University Press, Cambridge (2000)
3. Cheng, C.-C.: Word-focused extensive reading with guidance. In: Selected Papers from the Thirteenth International Symposium on English Teaching, pp. 24–32. Crane Publishing, Taipei (2004), `http://elearning.ling.sinica.edu.tw/` (last visited November 11, 2008)
4. Chu, B.-F.: Handbook of the Fifth Generation of the Cangjie Input Method, `http://www.cbflabs.com/book/ocj5/ocj5/index.html` (last visited on November 11, 2008)
5. Ezeiza, N., Maritxalar, M., Schulze, M. (eds.): Proc. of the Workshop on Natural Language Processing for Educational Resources. In: International Conference on Recent Advances in Natural Language Processing (2007)
6. Heift, T., Schulze, M.: Errors and Intelligence in Computer-Assisted Language Learning. Routledge, NY (2007)
7. Juang, D., Wang, J.-H., Lai, C.-Y., Hsieh, C.-C., Chien, L.-F., Ho, J.-M.: Resolving the unencoded character problem for Chinese digital libraries. In: Proceedings of the Fifth ACM/IEEE Joint Conference on Digital Libraries, pp. 311–319 (2005)
8. Lee, C.-Y.: Personal communication, Institute of Linguistics. Academia Sinica (2008)
9. Lee, H.: Cangjie Input Methods in 30 Days, Foruto Company, Hong Kong, `http://input.foruto.com/cjdict/Search_1.php` (last visited November 11, 2008)
10. Liu, C.-L., Wang, C.-H., Gao, Z.-M.: Using lexical constraints for enhancing computer-generated multiple-choice cloze items. International Journal of Computational Linguistics and Chinese Language Processing 10(3), 303–328 (2005)

11. Manning, C.D., Schüetze, H.: Foundations of Statistical Natural Language Processing. MIT Press, MA (1999)
12. Taft, M., Zhu, X., Peng, D.: Positional specificity of radicals in Chinese character recognition. Journal of Memory and Language 40, 498–519 (1999)
13. Tetreault, J., Burstein, J., De Felice, R. (eds.): Proc. of the Third Workshop on Innovative Use of NLP for Building Educational Applications, the Forty Sixth Annual Meeting of the Association for Computational Linguistics (2008)
14. Wang, F.F.Y.: Report on Chinese language concordances made by computer. Journal of the Chinese Language Teachers Association 1(2), 73–76 (1966)
15. Yeh, S.-L., Li, J.-L.: Role of structure and component in judgments of visual similarity of Chinese characters. Journal of Experimental Psychology: Human Perception and Performance 28(4), 933–947 (2002)
16. Zhang, Z.-S.: CALL for Chinese—Issues and practice. Journal of the Chinese Language Teachers Association 33(1), 51–82 (1998)
17. Tsay, Y.-C., Tsay, C.-C.: Diagnoses of Incorrect Chinese Usage. Firefly Publisher, Pan-Chiao, Taiwan (蔡有秩及蔡仲慶：新編錯別字門診，臺灣，板橋：螢火蟲出版社) (2003)

Adjusting Occurrence Probabilities of Automatically-Generated Abbreviated Words in Spoken Dialogue Systems

Masaki Katsumaru, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno

Graduate School of Informatics, Kyoto University, Kyoto, Japan
{katumaru,komatani,ogata,okuno}@kuis.kyoto-u.ac.jp

Abstract. Users often abbreviate long words when using spoken dialogue systems, which results in automatic speech recognition (ASR) errors. We define *abbreviated words* as sub-words of an original word and add them to the ASR dictionary. The first problem we face is that proper nouns cannot be correctly segmented by general morphological analyzers, although long and compound words need to be segmented in agglutinative languages such as Japanese. The second is that, as vocabulary size increases, adding many abbreviated words degrades the ASR accuracy. We have developed two methods, (1) to segment words by using conjunction probabilities between characters, and (2) to adjust occurrence probabilities of generated abbreviated words on the basis of the following two cues: phonological similarities between the abbreviated and original words and frequencies of abbreviated words in Web documents. Our method improves ASR accuracy by 34.9 points for utterances containing abbreviated words without degrading the accuracy for utterances containing original words.

Index Terms: Spoken dialogue systems, abbreviated words, adjusting occurrence probabilities.

1 Introduction

Users often omit parts of long words and utter abbreviated words [1]. For example, the abbreviated word *aoyamakan* (青山館), meaning Aoyama Hall, is said to indicate *aoyamaongakukinenkan* (青山音楽記念館), meaning Aoyama Memorial Hall of Music. They are apt to do this because users unfamiliar with a particular spoken dialogue system do not know much about how to use it and what content words are included in its vocabulary. In conventional system development, system developers manually add unknown words to an automatic speech recognition (ASR) dictionary by collecting and examining misrecognized words uttered by users. This manual maintenance requires a great deal of time and effort. Furthermore, a system cannot recognize these words until the manual maintenance has taken place. They continue to be misrecognized until the system developers find and add them to the system dictionary.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 481–490, 2009.

Our purpose is to automatically add abbreviated words users may utter at the initial time when an original dictionary in any domain has been provided. We define an *original dictionary* as the initial ASR dictionary for a system, *original words* as content words in an original dictionary, and *abbreviated words* as words that are sub-words of an original word and that indicate the same entity as the original word. We generate abbreviated words by omitting arbitrary sub-words of an original word. These abbreviated words are interpreted as their corresponding original words in a language understanding module. Automatic addition of vocabulary at the initial stage of system development alleviates manual maintenance time and effort. Furthermore, the system can recognize abbreviated words at an earlier stage, thus increasing its usability.

There are two problems when abbreviated words are added to an ASR dictionary.

1. **Segmenting proper nouns in order to generate abbreviated words** Proper nouns cannot be correctly segmented by general morphological analyzers because they are domain-dependent words, such as regional names. To decide which sub-words to omit, proper nouns need to be segmented in agglutinative languages such as Japanese, while words in an isolating language such as English do not pose this problem.
2. **Reducing ASR errors caused by adding abbreviated words to an ASR dictionary**
   ASR accuracy is often degraded by adding generated abbreviated words because the vocabulary size increases. Jan et al. merely added generated abbreviated words and did not take the degradation into account [2]. The following words tend to degrade ASR accuracy:
   (a) abbreviated words with phonemes close to those of other original words
   (b) abbreviated words that are not actually used

For the former, we segmented proper nouns by using conjunction probabilities between characters in addition to results of a morphological analyzer. For the latter, we manipulated occurrence probabilities of generated abbreviated words on the basis of phonological similarities between the abbreviated and original words [3]. We furthermore introduce a measure, *Web frequency*, for representing how much each generated abbreviated word is actually used. This measure is defined by using Web search results, and suppresses side effects caused by abbreviated words that are not used. These enable us to add abbreviated words to an ASR dictionary without increasing the ASR error rate.

2 Case Study of Deployed System

We preliminarily investigated gaps between users' utterances and the vocabulary of a system by analyzing words added by developers during the 5-year service of the Kyoto City Bus Information System [4]. Users stated their boarding stop as well as the destination or the bus route number by telephone, and the system informed them how long it would be before the bus arrived. There were 15,290 calls to the system during the 58 months between May 2002 and February 2007, and the system developers added users' words that the system could not recognize[1].

The developers added 309 words to the system's vocabulary. Of these 91.6% were aliases for already known entities, while 8.4% were new entities of bus stops and landmarks. There were far fewer new entities added than aliases for the already known entities. This means that the developers had carefully prepared the vocabulary for bus stops and landmarks at the initial stage of system development. The reason the added words consisted almost exclusively of aliases is that, at the initial stage of system development, the system developers were unable to predict the wide range of other expressions that would be uttered by real users. Abbreviated words were the majority of the added aliases, which were 78.3% of all added words. This means that real users actually often utter abbreviated words. Of the 1,494 utterances collected from novices using the system, 150 utterances contained abbreviated words.

3 Generating and Manipulating Occurrence Probabilities of Abbreviated Words

The flow of our method for adding abbreviated words is shown in Figure 1. First, original words are segmented to identify sub-words to omit. For domain-dependent proper nouns, a *conjunction probability* is defined between each character as a measure of segmenting compound words. As described in section 3.1, proper nouns are segmented by using conjunction probabilities and a morphological analyzer. Abbreviated words are then generated by omitting some sub-words of the segmented words.

In section 3.2, we address how to suppress ASR errors caused by adding generated abbreviated words. We define the phonological similarities between the abbreviated and original words, and the Web frequencies of the abbreviated words. Then occurrence probabilities are manipulated on the basis of them.

3.1 Segmenting Words in ASR Dictionary and Generating Abbreviated Words

In our method, a compound word in the ASR dictionary is first segmented into a sub-word array, "$s_1 s_2 \ldots s_n$". The segmentation is done at a part where either a morphological analyzer or conjunction probabilities would segment it. The morphological analyzer we use is MeCab [5]. Domain-dependent proper nouns are segmented by using conjunction probabilities between characters as follows. If a word in the ASR dictionary is expressed by the character string "$c_1 c_2 ... c_{i-1} c_i ... c_n$", a conjunction probability between c_{i-1} and c_i is formulated on the basis of the character N-gram probabilities in the ASR dictionary:

$$\min\{P(c_i|c_{i-1}c_{i-2}...c_1),\ \ P(c_{i-1}|c_i c_{i+1}\ldots c_n)\}. \tag{1}$$

[1] The developers did not add all words users uttered during this period. Short words were not added because they could cause insertion errors. This was because the system's dialogue management is executed in a mixed-initiated manner, and its language constraint is not so strong.

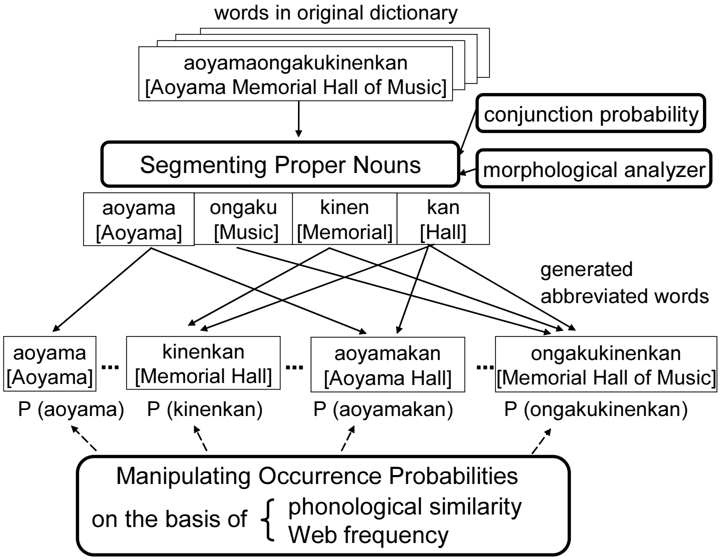


Fig. 1. Flow of adding abbreviated words

This means that a conjunction probability is defined as smaller one of N-gram probabilities forward to c_i and backward to c_{i-1}. A word is segmented between c_{i-1} and c_i if the conjunction probability between them is lower than threshold θ. For example, the proper noun *shisekikoenmae* (史跡公園前), which means “in front of the Historical Park”, is segmented as shown in Figure 2. Using conjunction probabilities segments *shisekikoenmae* (史跡公園前) into *shisekikoen* (史跡公園) and *mae* (前), while using MeCab cannot. This segmentation is essential to generating various abbreviated words such as *shisekikoen* (史跡公園).

Next, an arbitrary number of sub-words are omitted and $(2^n - 1)$ abbreviated words from a sub-word array “$s_1 s_2 \ldots s_n$” are generated. The pronunciations of the generated abbreviated words are given by the pronunciations of the sub-words, which are detected by matching the pronunciation given by MeCab and the original pronunciation.

3.2 Reducing ASR Errors Caused by Adding Generated Abbreviated Words

Definition of Phonological Similarity. We define phonological similarity as a measure of confusion in ASR that is caused by generated abbreviated words. These words may cause ASR errors for utterances containing original words when the phonemes of the generated words are close to those of the original words or those of parts of the original words. We define the phonological similarity between generated abbreviated word w and vocabulary D_{org} of the original dictionary as

$$dist(w, D_{org}) = \min(\ e.d.(w, part(D'_{org}))\ ). \qquad (2)$$

Segmentation result of MeCab

shiseki [Historical]	koenmae [in front of Park]

Segmentation result of conjunction probabilities

shisekikoen [Historical Park]	mae [in front of]

↓

Segmentation result of MeCab and conjunction probabilities

shiseki [Historical]	koen [Park]	mae [in front of]

Fig. 2. Segmentation result of "shisekikoenmae"

We denote D'_{org} as the vocabulary made by removing from D_{org} words from which w is generated. Partial sequences of all words of D'_{org} are given by $part(D'_{org})$. The edit distance between x's and y's phoneme strings is $e.d.(x, y)$; it is calculated by DP matching [6]. If we define S_1 as a phoneme set of vowels, a moraic obstruent and a moraic nasal, and S_2 as a phoneme set of consonants, we set costs of edit distance 2 when an element of S_1 is inserted, deleted, or substituted, and 1 when an element of S_2 is inserted, deleted, or substituted with one of S_2.

Definition of Web Frequency. We define the Web frequency of a generated abbreviated word as the frequency it appears in Web documents. The Web frequencies of words indicate how often they are actually used. The frequency is obtained by performing a query on a Web search engine. We used Yahoo! Japan[2]. We define the Web frequency of a generated abbreviated word as

$$WebFrequency(w) = \frac{count(w)}{count(original(w))}, \tag{3}$$

in which $count(\text{``}word\text{''})$ is the hit count of Web pages for query "*word*", and $original(w)$ is the original word from which w was generated. We normalize $count(w)$ by $count(original(w))$ to give proper measures because $count(w)$ tends to be small (or high) when $count(original(w))$ is small (or high). The lower the Web frequency, the less frequently users may utter.

We generated abbreviated words from the vocabulary of the Kyoto City Bus Information System. The phonological similarities and Web frequencies of some of these words are shown in Table 1. The phonological similarity between the generated abbreviated word *horikawashi* and the vocabulary of the original dictionary is 0 because *horikawashi* is equal to part of *horikawashimodachiuri*. The similarity of *rokuhamitsuji* is same as that of *paresusaido*. However we should set the probability of *rokuhamitsuji* differently from that of *paresusaido* because *rokuhamitsuji* is generated by a segmentation error and not actually used, but *paresusaido* is used. The Web frequency of *rokuhamitsuji* is much lower than that

[2] http://search.yahoo.co.jp

Table 1. Phonological similarities and Web frequencies of generated abbreviated words

Abbreviated word (its original word)	*P.S.*	Closest original word	*W.F.*
horikawashi (horikawatakoyakushi) [Name of Area]	0	horikawashimodachiuri [Name of Area]	$5.77 * 10^{-2}$
shakadani (shakadaniguchi) [Name of Area]	2	haradani [Name of Area]	$2.75 * 10$
rokuhamitsuji (rokuharamitsuji) [Name of Temple]	6	kokuritsukindaibijutsukan [Name of Museum]	0.00
paresusaido (kyotoparesusaidohoteru) [Name of Hotel]	6	karasumashimochojamachi [Name of Town]	$3.86 * 10^2$

P.S.: phonological similarity
W.F.: Web frequency

of *paresusaido*. We can thus distinguish between *rokuhamitsuji* and *paresusaido* by considering Web frequency.

Manipulating Occurrence Probabilities on the basis of Phonological Similarity and Web Frequency. Degradation of ASR accuracy for utterances containing original words is avoided by manipulating the occurrence probabilities of the generated abbreviated words on the basis of their Web frequencies in addition to their phonological similarities. We define $P_{org}(w)$ as the occurrence probability of word w. The probabilities of the generated abbreviated words that meet two conditions,

$$dist(w, D_{org}) \leq d \tag{4}$$

$$WebFrequency(w) \leq e \tag{5}$$

$$(d,\ e:\ threshold)$$

are arranged as new occurrence probabilities:

$$P_{new}(w) = P_{org}(w) * \alpha^{dist(w,D_{org})-d-1} * WebFrequency(w). \tag{6}$$

Generated abbreviated words that meet only (4) are arranged as

$$P_{new}(w) = P_{org}(w) * \alpha^{dist(w,D_{org})-d-1}, \tag{7}$$

and those that meet only (5) are arranged as

$$P_{new}(w) = P_{org}(w) * WebFrequency(w). \tag{8}$$

We set α to 10. The lower the phonological similarity and Web frequency, the lower the occurrence probability. Generated abbreviated words with a Web frequency of 0 are removed from the ASR dictionary. $P_{new}(w)$ is calculated for all generated abbreviated words. We then normalize the probabilities of the original and generated abbreviated words to meet $\sum_{word \in W} P(word) = 1$, in which W is a set of all the original and abbreviated words.

4 Experimental Evaluation

We experimentally evaluated our method. We generated abbreviated words from a system's ASR dictionary and added them to the dictionary. The metrics were the recall rate and the ASR accuracy for the collected utterances. The ASR accuracy is calculated as $(Cor - Ins)/Len$ * 100 [%], in which Cor, Ins, and Len are the number of correction, insertion, and words in the manual transcription. To verify whether our method is independent of a particular domain, we also generated abbreviated words in another domain.

4.1 Target Data for Evaluation

We used real users' utterances collected on the Kyoto City Bus Information System. We targeted users who were not familiar with the system's vocabulary and collected utterances of users who were using the system for the first time by analyzing their telephone numbers. We collected 1,494 utterances by 183 users after removing utterances that were not relevant to the task. Of the 1,494 utterances, 150 contained 70 kinds of abbreviated words, and 1,142 contained only original words. The other 202 consisted of words that were neither abbreviated nor original such as, "Can this system tell me where to change buses?".

4.2 Recall Rate of Generated Abbreviated Words

We generated abbreviated words from 1,481 words (bus stops and landmarks) of the 1,668 original words in the Kyoto Bus Information System dictionary. The threshold θ of segmentation by conjunction probabilities was set at 0.12 after preliminary experiments. We segmented the words by using both conjunction probabilities and MeCab, omitted sub-words, and generated 11,936 abbreviated words. To prove the effectiveness of our segmentation, we also generated 2,619 abbreviated words by segmentation using only conjunction probabilities and 8,941 using only MeCab. We evaluated three methods of segmentation:

- conjunction probabilities only
- MeCab (morphological analyzer) only
- both conjunction probabilities and MeCab (our method).

The recall rates for abbreviated words generated by each method are shown in Table 2. For 70 different abbreviated words uttered by real users in the collected data, our method generated 66 (94%) while 51 (73%) were generated by only conjunction probabilities and 60 (86%) by using only MeCab. The recall rate with our method was 8 points higher than that with only MeCab. Using conjunction probabilities led to this improvement.

4.3 Evaluation of ASR Accuracy

We constructed a statistical language model in order to manipulate the occurrence probabilities for each word because the Kyoto City Bus Information System's ASR is a grammar-based one. First, content words were assigned to

Table 2. Recall rate for each segmentation method

Method of segmentation	Number of generated abbreviated words	Recall rate [%]
conjunction probabilities only	2,619	73
MeCab (morphological analyzer) only	8,941	86
MeCab + conjunction probabilities (our method)	11,936	94

the class of bus stops, landmarks, and bus route numbers. Next, we constructed a class N-gram model from all kinds of sentences that the grammar-based language model generated. We used a CMU Toolkit [7] to construct the statistical language model. We added the abbreviated words generated to the class of bus stops and landmarks in addition to original bus stops or landmarks. The acoustic model was a triphone model with 2,000 states and 16 mixture components for telephone speech. The ASR engine was Julius [8]. We set d to 5 and e to 400,000 by trial and error. The following are the experimental conditions:

Cond. 1: original dictionary (baseline)
Use the system's original ASR dictionary before adding abbreviated words (vocabulary size: 1,668)

Cond. 2: Cond. 1 + generated abbreviated words
Add generated abbreviated words to original dictionary (13,604)

Cond. 3: Cond. 2 + manipulating occurrence probabilities on the basis of only phonological similarity
Add generated abbreviated words to original dictionary and manipulate occurrence probabilities on the basis of only phonological similarity (13,604)

Cond. 4: Cond. 2 + manipulating occurrence probabilities on the basis of only Web frequency
Add generated abbreviated words to the original dictionary and manipulate occurrence probabilities on the basis of only Web frequency (7,203)

Cond. 5: Cond. 2 + manipulating occurrence probabilities on the basis of both phonological similarity and Web frequency (our method)
Add generated abbreviated words to the original dictionary and manipulate occurrence probabilities on the basis of both phonological similarity and Web frequency (7,203)

Table 3 shows the ASR accuracy of content words for 150 utterances with abbreviated words, 1,142 utterances with only original words, and all 1,494 utterances.

Comparing Cond. 1 and 2, the ASR accuracy for all utterances in Cond. 2 degraded by 12.3 points although that for utterances with abbreviated words in Cond. 2 improved by 23.6 points. This is because we only added abbreviated words generated by our method. This result shows that ASR accuracy degrades by merely adding these words. In Cond. 3, the ASR accuracy for utterances with original words improved by 15.0 points compared with Cond. 2 and degraded only 0.1 points compared with Cond. 1. This came from manipulating the probabilities of generated abbreviated words on the basis of phonological similarities. This result shows that phonological similarity based manipulation reduces ASR

Table 3. ASR accuracy [%] for content words of utterances for each condition

Condition	Utterances with abbreviated words	Utterances with original words	All utterances
1: original dictionary (baseline)	1.1	74.9	52.5
2: 1 + generated abbr.	24.7	59.8	40.2
3: 2 + (a) phonological similarity	25.3	74.8	56.5
4: 2 + (b) Web frequency	25.3	44.5	27.6
5: 2 + (a) + (b) (our method)	36.0	76.0	58.9

errors caused by adding abbreviated words. Comparing Cond. 2 and 4, the ASR accuracy for utterances with abbreviated words in Cond. 4 was slightly higher. This is because we arranged probabilities on the basis of Web frequency. This indicates that Web frequency based manipulation of the probabilities suppresses ASR errors caused by generated abbreviated words not actually used. However the ASR accuracy for all utterances in Cond. 4 was degraded compared with the accuracy in Cond. 2. This was because high occurrence probabilities were given to short words that frequently appeared in the Web, and accordingly insertion errors increased. In Cond. 5, the ASR accuracy for utterances with abbreviated words increased by 34.9 points compared with that in Cond. 1, and increased by 10.7 points compared with that in Cond. 3 or 4. The ASR accuracy for utterances with original words in Cond. 5 did not degrade when compared with Cond. 1. This was because we used both phonological similarity and Web frequency to adjust occurrence probabilities. These results demonstrate the effectiveness of our method of manipulating occurrence probabilities on the basis of both phonological similarities and Web frequencies for reducing ASR errors caused by adding abbreviated words.

The ASR accuracy is still low throughout the experiment. A reason for this low level of accuracy is a mismatch between the acoustic model and the users' circumstances. Actually, there were several cases in which acoustic scores for correct word sequences were lower than those for others. We have addressed how to improve the language model. Improving the acoustic model will lead to a higher level of ASR accuracy.

4.4 Generating Abbreviated Words in Another Domain

We also generated abbreviated words for the restaurant domain to verify whether our method is independent of a particular domain. We check only generated abbreviated words because we have no dialogue data in this domain and cannot evaluate ASR accuracy. In this domain as well, domain-dependent proper nouns were correctly segmented by using conjunction probabilities, and several appropriate abbreviated words were generated although the morphological analyzer could not segment some of them. For example, our method could segment *bisutorokyatorudoru* into *bisutoro (bistro)* and *kyatorudoru (name of restaurant)* by detecting the high frequency of *bisutoro* in the dictionary, although MeCab could not segment it. This segmentation enabled us to generate the abbreviated word *kyatorudoru*, which is often used.

5 Conclusion

We generated abbreviated words and added them to an ASR dictionary to enable a dialogue system to recognize abbreviated words uttered by users. To increase the recall rate of the generated abbreviated words, we segment proper nouns by introducing conjunction probabilities between characters in the system's dictionary. To add abbreviated words without increasing the ASR error rate, we manipulate their occurrence probabilities on the basis of their Web frequency (the frequency of their use in Web documents) in addition to the phonological similarity between the abbreviated and original words.

Experimental evaluations using real users' utterances demonstrated that our method is effective. The recall rate was higher than that using only a morphological analyzer. The ASR accuracy for utterances with abbreviated words was 34.9 points higher than that when only the original dictionary was used without degrading the accuracy for utterances with original words. These results show that our method for vocabulary expansion enables a dialogue system to recognize user's abbreviated words without increasing the ASR error rate. Future work includes collecting utterances in another domain and using them to evaluate our method.

Acknowledgments. We are grateful to Dr. Shun Shiramatsu of Kyoto University for allowing us to use the Web page counting program he developed.

References

1. Zweig, G., Nguyen, P., Ju, Y., Wang, Y., Yu, D., Acero, A.: The Voice-Rate Dialog System for Consumer Ratings. In: Proc. Interspeech, pp. 2713–2716 (2007)
2. Jan, E.E., Maison, B., Mangu, L., Zweig, G.: Automatic Construction of Unique Signatures and Confusable Sets for Natural Language Directory Assistance Applications. In: Proc. Eurospeech, pp. 1249–1252 (2003)
3. Katsumaru, M., Komatani, K., Ogata, T., Okuno, H.G.: Expanding Vocabulary for Recognizing User's Abbreviations of Proper Nouns without Increasing ASR Error Rates in Spoken Dialogue Systems. In: Proc. Interspeech, pp. 187–190 (2008)
4. Komatani, K., Ueno, S., Kawahara, T., Okuno, H.G.: User Modeling in Spoken Dialogue Systems for Flexible Guidance Generation. In: Proc. Eurospeech, pp. 745–748 (2003)
5. Kudo, T., Yamamoto, K., Matsumoto, Y.: Applying conditional random fields to Japanese morphological analysis. In: Proc. EMNLP, pp. 230–237 (2004), `http://mecab.sourceforge.net/`
6. Navarro, G.: A Guided Tour to Approximate String Matching. ACM Computing Surveys 33(1), 31–88 (2001)
7. Clarkson, P.R., Rosenfeld, R.: Statistical Language Modeling Using the CMU-Cambridge Toolkit. In: Proc. ESCA Eurospeech, pp. 2707–2710 (1997), `http://svr-www.eng.cam.ac.uk/~prc14/toolkit.html`
8. Kawahara, T., Lee, A., Takeda, K., Itou, K., Shikano, K.: Recent progress of open-source LVCSR Engine Julius and Japanese model repository. In: Proc. ICSLP, pp. 3069–3072 (2004)

Accelerating Collapsed Variational Bayesian Inference for Latent Dirichlet Allocation with Nvidia CUDA Compatible Devices

Tomonari Masada, Tsuyoshi Hamada, Yuichiro Shibata, and Kiyoshi Oguri

Nagasaki University
Bunkyo-machi 1-14, Nagasaki, Japan
{masada,hamada,shibata,oguri}@cis.nagasaki-u.ac.jp

Abstract. In this paper, we propose an acceleration of collapsed variational Bayesian (CVB) inference for latent Dirichlet allocation (LDA) by using Nvidia CUDA compatible devices. While LDA is an efficient Bayesian multi-topic document model, it requires complicated computations for parameter estimation in comparison with other simpler document models, e.g. probabilistic latent semantic indexing, etc. Therefore, we accelerate CVB inference, an efficient deterministic inference method for LDA, with Nvidia CUDA. In the evaluation experiments, we used a set of 50,000 documents and a set of 10,000 images. We could obtain inference results comparable to sequential CVB inference.

1 Introduction

In this paper, we present an application of general-purpose GPU to parameter inference for probabilistic document models. We accelerate an inference method, called *collapsed variational Bayesian (CVB) inference* [12], for a well-known Bayesian multi-topic document model, *latent Dirichlet allocation (LDA)* [3], by using Nvidia *compute unified device architecture (CUDA)* [1] compatible devices. First of all, we summerize our two main contributions. Our research is the first attempt to parallelize *CVB inference*, and also the first attempt to parallelize inference for LDA *by using GPU*. The rest of the paper is organized as follows. Section 2 includes the background of our research. Section 3 presents the details of parallelized CVB inference for LDA. Section 4 shows how to implement parallelized CVB inference by using Nvidia CUDA compatible devices. Section 5 provides the results of our experiments. Section 6 concludes the paper.

2 Background

2.1 Latent Dirichlet Allocation

LDA [3] is a *Bayesian multi-topic* document model. The term *Bayesian* refers to probabilistic models where, after introducing prior distributions, posterior distribution is estimated not by a specific set of parameter values, but as a

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 491–500, 2009.

distribution over all possible parameter values. The meaning of the term *multi-topic* can be explained as follows. In document modeling, topic is often identified with a multinomial distribution defined over words, because semantic differences are reflected in what kind of words are frequently used. In LDA, a set of words constituting one document are drawn from more than one multinomials. Namely, documents are modeled as a mixture of *multiple topics*.

While LDA is originally proposed as a probabilistic model of documents, we can find its applications in various research fields [13][14]. However, due to complicated model structure, LDA requires acceleration when applied to datasets of large size. We can apply Expectation-Maximization (EM) algorithm to simpler document models, e.g. Dirichlet compound multinomial [7] and probabilistic latent semantic indexing [6]. EM algorithm can be efficiently implemented in a parallelized manner [4]. In contrast, we cannot use EM for LDA. Therefore, we propose an acceleration customized for LDA by using general-purpose GPU.

2.2 Collapsed Variational Bayesian Inference

The following three inference methods are so far proposed for LDA: variational Bayesian (VB) inference [3], collapsed Gibbs sampling [5], and CVB inference. In this paper, we focus on CVB inference due to the following two reasons. First, both VB and CVB inferences use variational method to obtain a tractable posterior distribution. While variational method introduces approximation, CVB inference can achieve less approximation than VB for LDA [12]. Second, collapsed Gibbs sampling incorporates randomization. Therefore, we should carefully determine when to stop inference iteration. In contrast, CVB and VB inferences are deterministic methods. It is relatively easy to decide when to terminate iteration.

Among these three inference methods, VB inference and collapsed Gibbs sampling have already been parallelized by using PC clusters [8][11][10]. However, both methods divide a given dataset into smaller subsets and process the subsets in parallel. As far as intermediate results are appropriately broadcasted in the course of inference, we can adopt this *coarse-grained* parallelization only by introducing negligible approximations. Our research focuses on another parallelism appearing in parameter update formula of CVB inference. Our method is based on *fine-grained* parallelism, whose details are outlined below.

Parameters to be estimated in CVB inference for LDA are indexed by documents, words, and topics. Let $j = 1, \ldots, J$, $w = 1, \ldots, W$, and $k = 1, \ldots, K$ be the indexes of documents, words, and topics, respectively. Let γ_{jwk} denote parameters to be estimated for all document/word pairs j, w and for all topics k. Intuitively speaking, γ_{jwk} means how strongly word w in document j relates to topic k. An outline of one iteration, i.e. one dataset scan, of CVB inference for LDA is shown in Fig. 1. When the number of unique document/word pairs is M, time complexity of each iteration is $O(MK)$.

Our fine-grained parallelism originates from the fact that parameters corresponding to different topics can be updated independently. Namely, all values required for updating parameters indexed by a specific k are also indexed by the same k. Therefore, we can conduct K update computations in parallel, where K

```
for each document $j$ in a given document set
    for each word $w$ appearing in document $j$
        for each topic $k$
            update $\gamma_{jwk}$
        next
        normalize $\gamma_{jw1}, \dots, \gamma_{jwK}$ so that $\sum_{k=1}^{K} \gamma_{jwk} = 1$ holds
    next
next
```

Fig. 1. An outline of one iteration of CVB inference for LDA

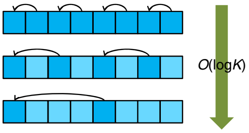


Fig. 2. A standard parallel reduction for parameter normalization

is the number of topics. This is the fine-grained parallelism used by our method. However, $\sum_k \gamma_{jwk} = 1$ should hold for each document/word pair j, w. This normalization can be realized by a standard reduction (See Fig. 2) of $O(\log K)$ time. Therefore, time complexity of each iteration is reduced to $O(M \log K)$. Further, our fine-grained parallelism is orthogonal to coarse-grained parallelism, and thus both types of parallelism can be combined.

Recently, a new acceleration method of collapsed Gibbs sampling for LDA appears [9]. This method reduces time complexity by an algorithmic elaboration. Interestingly, our method shares the same intuition, because both methods try to reduce $O(K)$ factor of time complexity for LDA inferences.

3 Details of Parallelized Inference

3.1 Task of CVB Inference for LDA

We can describe LDA document model as a process of generating documents. First, we draw a topic multinomial distribution $Mul(\theta_j)$ for each document j from a symmetric Dirichlet prior distribution $Dir(\alpha)$. Second, we draw a word multinomial $Mul(\phi_k)$ for each topic k from another symmetric Dirichlet prior $Dir(\beta)$. Then, we can generate document j by repeating the following procedure as many times as the number of word tokens in document j: draw a topic from $Mul(\theta_j)$, and then draw a word from $Mul(\phi_k)$ when the drawn topic is k. This generative description leads to full joint distribution as follows:

$$p(\mathbf{x}, \mathbf{z}, \theta, \phi; \alpha, \beta) = \prod_k p(\phi_k; \alpha) \prod_j p(\theta_j; \beta) \cdot \prod_j \prod_i p(z_{ji}|\theta_j) p(x_{ji}|\phi_{z_{ij}}) \quad (1)$$

where x_{ji} is the random variable whose value is ith word token in document j, and z_{ji} is the random variable whose value is the topic used to draw ith word token in document j. In this paper, we say that topic k *is assigned to* a certain word token when the word token is drawn from a word multinomial $Mul(\phi_k)$.

From Eq. (1), we obtain posterior distribution $p(\mathbf{z}, \theta, \phi|\mathbf{x}; \alpha, \beta)$ as $p(\mathbf{x}, \mathbf{z}, \theta, \phi; \alpha, \beta) \,/p(\mathbf{x}; \alpha, \beta)$. However, marginal likelihood $p(\mathbf{x}; \alpha, \beta)$ is intractable. Therefore, we approximate posterior by variational method. Let $q(\mathbf{z}, \theta, \phi)$ denote an approximated posterior. Then, log marginal likelihood can be lower bounded as:

$$\log p(\mathbf{x}; \alpha, \beta) \geq \int \int \sum_{\mathbf{z}} q(\mathbf{z}, \theta, \phi) \log \frac{p(\mathbf{x}, \mathbf{z}, \theta, \phi; \alpha, \beta)}{q(\mathbf{z}, \theta, \phi)} d\theta d\phi \tag{2}$$

In CVB inference, we assume that topic assignments z_{ji} are mutually independent and approximate posterior as:

$$q(\mathbf{z}, \theta, \phi) = p(\theta, \phi|\mathbf{z}, \mathbf{x}, \alpha, \beta)q(\mathbf{z}) = p(\theta, \phi|\mathbf{z}, \mathbf{x}, \alpha, \beta) \prod_j \prod_i q(z_{ji}; \gamma_{jx_{ji}}) \tag{3}$$

where $q(z_{ji}; \gamma_{jx_{ji}})$ is an approximated posterior probability of topic assignment to ith word in document j. A set of K parameters $\gamma_{jw} = \{\gamma_{jw1}, \ldots, \gamma_{jwK}\}$ for a fixed document/word pair j, w can be regarded as a topic multinomial distribution. γ_{jwk} means how strongly word w in document j relates to topic k. By using Eq. (3), the right hand side of Eq. (2) is reduced to $\sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}; \alpha, \beta)}{q(\mathbf{z})}$, where all model parameters θ, ϕ are marginalized out. This marginalization is denoted by the term *collapsed*. The task of CVB inference is to determine posterior parameters by maximizing $\sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}; \alpha, \beta)}{q(\mathbf{z})}$.

3.2 Parameter Update Formula

We use CVB inference accompanied with Gaussian approximation presented in [12]. In this paper, we only show resulting update formula. Three types of mean/variance pairs, defined below, are needed to update posterior parameters:

$$\begin{aligned} M_{jk} &\equiv \sum_w n_{jw}\gamma_{jkw}, & V_{jk} &\equiv \sum_w n_{jw}\gamma_{jkw}(1 - \gamma_{jkw}) \\ M_{kw} &\equiv \sum_j n_{jw}\gamma_{jkw}, & V_{kw} &\equiv \sum_j n_{jw}\gamma_{jkw}(1 - \gamma_{jkw}) \\ M_k &\equiv \sum_{j,w} n_{jw}\gamma_{jkw}, & V_k &\equiv \sum_{j,w} n_{jw}\gamma_{jkw}(1 - \gamma_{jkw}) \end{aligned} \tag{4}$$

where n_{jw} is the number of tokens of word w in document j. By using these three types of mean/variance pairs, posterior parameters are updated as shown in Fig. 3, which is a detailed description of the innermost loop of Fig. 1. Fig. 3 shows that update computations for different ks can be executed in parallel. Only normalization in Step 3 requires $O(\log K)$ time based on a standard reduction technique in Fig. 2. Therefore, time complexity per iteration is $O(M \log K)$.

1. Subtract the contribution of γ_{jwk} from all three types of means and variances.

$$\begin{aligned} M_{jk} &\leftarrow M_{jk} - n_{jw}\gamma_{jwk}, & V_{jk} &\leftarrow V_{jk} - n_{jw}\gamma_{jwk}(1-\gamma_{jwk}) \\ M_{kw} &\leftarrow M_{kw} - n_{jw}\gamma_{jwk}, & V_{kw} &\leftarrow V_{kw} - n_{jw}\gamma_{jwk}(1-\gamma_{jwk}) \\ M_{k} &\leftarrow M_{k} - n_{jw}\gamma_{jwk}, & V_{k} &\leftarrow V_{k} - n_{jw}\gamma_{jwk}(1-\gamma_{jwk}) \end{aligned}$$

2. Update γ_{jwk}.

$$\begin{aligned} \gamma_{jwk} \leftarrow & (\alpha + M_{jk})(\beta + M_{kw})(W\beta + M_k)^{-1} \\ & \cdot \exp\{-\frac{V_{jk}}{2(\alpha+M_{jk})^2} - \frac{V_{kw}}{2(\beta+M_{kw})^2} + \frac{V_k}{2(W\beta+M_k)^2}\} \end{aligned}$$

3. Normalize $\gamma_{jw1}, \ldots, \gamma_{jwK}$ so that $\sum_k \gamma_{jwk} = 1$ holds.
4. Add the contribution of γ_{jwk} to all three types of means and variances.

$$\begin{aligned} M_{jk} &\leftarrow M_{jk} + n_{jw}\gamma_{jwk}, & V_{jk} &\leftarrow V_{jk} + n_{jw}\gamma_{jwk}(1-\gamma_{jwk}) \\ M_{kw} &\leftarrow M_{kw} + n_{jw}\gamma_{jwk}, & V_{kw} &\leftarrow V_{kw} + n_{jw}\gamma_{jwk}(1-\gamma_{jwk}) \\ M_{k} &\leftarrow M_{k} + n_{jw}\gamma_{jwk}, & V_{k} &\leftarrow V_{k} + n_{jw}\gamma_{jwk}(1-\gamma_{jwk}) \end{aligned}$$

Fig. 3. How to update parameters in CVB inference

4 Implementation on CUDA

4.1 Nvidia CUDA Compatible Devices

Fig. 4 shows execution model of Nvidia CUDA compatible GPU devices. We only describe necessary details. The largest execution unit is called *grid*. In our case, one grid roughly corresponds to one iteration, i.e., one scan of dataset. However, when dataset is too large for graphics card RAM, called *device memory*, we divide dataset into subsets and process the subsets sequentially. In this case, one grid corresponds to processing of one data subset. Typical high-end graphics cards provide up to 1 GBytes device memory.

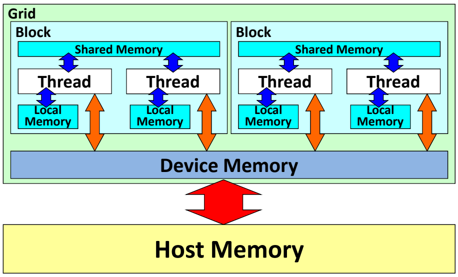


Fig. 4. Conceptual digram of Nvidia CUDA compatible devices

Each grid contains hundreds of *blocks*, each of which corresponds to one document in our case. Therefore, we process as many documents as blocks in parallel. Each block includes hundreds of *threads*. Threads in the same block can communicate with each other by using a fast memory, called *shared memory*. Shared memory is as fast as *local memory* assigned to each thread. Only four clock cycles are required to both read from and write to shared memory or local memory. However, threads from different blocks should communicate via device memory, which requires hundreds of clock cycles to access. Further, shared memory size is only tens of KBytes per block in existing graphics cards. Therefore, efficient communication among threads can be realized only for small size data.

A typical usage of CUDA compatible devices is as follows: send data from host memory (i.e., CPU memory) to device memory, execute computations on GPU, and then write results back from device memory to host memory. Since data transfer between host memory and device memory requires quite a large number of clock cycles, threads should not access to host memory during computation. Therefore, we send data to device memory before launching a grid on GPU.

4.2 Implementation Details of CVB Inference for LDA

By considering features of CUDA described above, we implement a parallelized version of CVB inference for LDA. Different threads are responsible for computations relating to different topics. This is our fine-grained parallelization. We assume that K is less than the maximum number of threads per block. This assumption is admissible for most applications.

We have following variables in CVB inference: posterior parameters γ_{jwk} and three types of mean/variance pairs (M_{jk}, V_{jk}), (M_{kw}, V_{kw}), (M_k, V_k) as shown in Fig. 3. The number of posterior parameters is too large for shared memory. Therefore, we store the parameters on device memory. However, the same set of K parameters $\gamma_{jw1}, \ldots, \gamma_{jwK}$ are used for four consecutive steps in Fig. 3 with respect to a fixed document/word pair. This means that, before executing each sequence of these four steps, we can load the corresponding set of K parameters from device memory to shared memory. Normalization can also be efficiently conducted on shared memory.

We store the first type of mean/variance pairs (M_{jk}, V_{jk}) on local memory. There are three reasons. First, this type of pairs is not shared by threads from different blocks, because different blocks process different documents. Second, both local memory and shared memory are large enough to store all K mean/variance pairs (M_{j1}, V_{j1}), ..., (M_{jK}, V_{jK}) for a fixed j. Third, these pairs do not need to be shared among different threads even in the same block.

The second type of mean/variance pairs (M_{kw}, V_{kw}) should be shared by the threads processing the same word. However, the same word can appear in many different documents. Therefore, we store these mean/variance pairs on device memory. Further, multiple threads may access the same pair at the same moment when they occasionally process the same word simultaneously. Mutual exclusion is required here, because all mean/variance pairs are not only read, but also modified, as shown in Fig. 3. Preliminary experiments show that atomic functions prepared for CUDA largely increase execution time. Therefore, we do not implement mutual exclusion. This leads to an introduction of approximation, because non-atomic read and write operations from different threads are not ordered appropriately. Further, some of these operations may fail. Preliminary experiments show that this approximation is negligible as long as the number of blocks per grid is small enough. Based on our experiences, the number of blocks per grid should be at most 16. Therefore, at most $512 \times 16 = 8192$ threads run in parallel. With larger number of threads, inference will not proceed correctly.

The third type of mean/variance pairs (M_k, V_k) should also be shared by threads from different blocks. However, we store this type of mean/variance pairs on local memory and reduce access to device memory. As a result, different blocks use different values. However, this type of mean/variance pairs is only indexed by topics. Namely, these means and variances are obtained by summation over document/word pairs. We can expect that only negligible discrepancy will be observed among this type of mean/variance pairs localized to different blocks.

Only posterior parameters are written back from device memory to host memory after a launch of a grid. Three types of means and variances are computed based on Eq. (4) from scratch on CPU by using all posterior parameters. In many realistic applications, we will divide an input dataset into smaller subsets and process them on GPU sequentially, because device memory is not large enough. Therefore, a single launch of grid updates only a small part of posterior parameters. When we compute means and variances by using all posterior parameters including those which are not updated in the preceding launch of grid, approximations introduced into our parallelized inference will be reduced.

5 Experiments

5.1 Settings

In this paper, we evaluate inference quality by *test data perplexity*. We use one half of the tokens of each word in each document as training data for CVB inference. Another half is used as test data for perplexity evaluation. Test data perplexity is computed based on test data probability which is defined as $p(\mathbf{x}_{test}) = \prod_j \prod_w \left(\sum_k \frac{\alpha + M_{jk}}{K\alpha + \sum_k M_{jk}} \frac{\beta + M_{kw}}{W\beta + M_k} \right)^{n'_{jw}}$, where n'_{jw} is the number of test data tokens of word w in document j. Then, test data perplexity can be defined as $\exp\{-\log p(\mathbf{x}_{test}) / \sum_{j,w} n'_{jw}\}$. After a large enough number of iterations, test data perplexity reaches a minimum, which is not guaranteed to be a global minimum in CVB inference. We evaluate inference quality by test data perplexity after convergence. When our parallelized CVB inference gives test data perplexities comparable to sequential version, our parallelization is admissible.

All experiments are conducted on a PC equipped with Intel Core2 Quad CPU Q9550 at 2.83GHz and with 8 GBytes host memory. As an Nvidia CUDA compatible device, we used Leadtek WinFast GTX 260 with 896 MBytes device memory, where shared memory size per block is 16 KBytes, maximum number of threads per block is 512, and clock rate is 1.24 GHz. Based on preliminary experiments, we set the number of blocks in a grid to 16. With larger number of blocks per grid, inference will not proceed correctly due to too many simultaneous non-atomic accesses to device memory as described in Section 4.2. The number of threads per block is set to 512. Therefore, $512 \times 16 = 8192$ threads can run in parallel. With this setting, 20 GFLOPS was achieved for computations on GPU in our experiments. When K is far less than the number of threads per block, we can assign more than one documents to each block. For example, when $K = 64$, we can process $512/64 = 8$ documents in parallel per block.

5.2 Datasets

We used a document set and an image set for comparison experiments. The document set consists of 56,755 Japanese newswire articles from Mainichi and Asahi newspaper Web sites. Dates of articles range from November 16, 2007 to May 15, 2008. By using morphological analyzer *MeCab* [2], we obtain 5,053,978 unique document/word pairs. The number of word tokens is 7,494,242, among which 1,666,156 are used for perplexity evaluation. The number of words is 40,355 after removing rare words and stop words. For this dataset, we tested two settings for the numbers of topics, i.e., $K = 64$ and $K = 128$.

The image set is *10,000 test images* by Professor J.-Z. Wang [16] [15]. Since our aim is not to propose a new image processing method, we used a simple feature extraction. Regardless image size, we divide each image into $16 \times 16 = 256$ non-overlapping rectangle regions of the same width and height. Further, we uniformly quantize RGB intensities and reduce the number of colors from 256^3 to $8^3 = 512$. Then, we count the frequency of quantized colors in each of 256 rectangle regions. Consequently, all images are represented as a frequency distribution defined over $256 \times 512 = 131,072$ features. After removing features appearing too frequently, we obtain $113,181$ features, which is the number of words in LDA. Exact number of images in this dataset, i.e., the number of documents, is 9,908. The number of unique document/word pair is 17,127,235, and the number of tokens is 84,741,744, among which 40,941,092 tokens are used for perplexity evaluation. For this dataset, we set the number of topics to 64.

5.3 Results

We run 10 trials of both parallel and sequential CVB inferences for LDA starting from randomly initialized posterior parameters. We present test data perplexities for the first 64 iterations, because further iterations give no significant changes. Table 1 provides test data perplexity and execution time after 64 iterations. Further, Fig. 5 shows test data perplexity versus execution time.

Space complexity scales with K in CVB inference for LDA. Therefore, we need more frequent data transfers between host memory and device memory for larger K. Table 1 shows that we could only achieve ×4.6 acceleration for $K = 128$ in comparison with ×7.3 for $K = 64$ with respect to the same document set. However, CUDA could provide comparable perplexities for our document set. For Professor Wang's image set, we even achieved less perplexities,

Table 1. Test data perplexity and execution time after 64 iterations. Mean values for 10 trials are presented. We also show standard deviation for test data perplexity.

	document set, $K = 64$	document set, $K = 128$	image set, $K = 64$
without CUDA	1300.0 ± 10.7 (2h 06m 56s)	1001.0 ± 9.3 (4h 12m 49s)	5604.1 ± 30.3 (7h 10m 24s)
with CUDA	1295.2 ± 17.2 (17m 19s; ×7.3)	1005.4 ± 8.7 (54m 25s; ×4.6)	5015.4 ± 38.2 (2h 30m 50s; ×2.9)

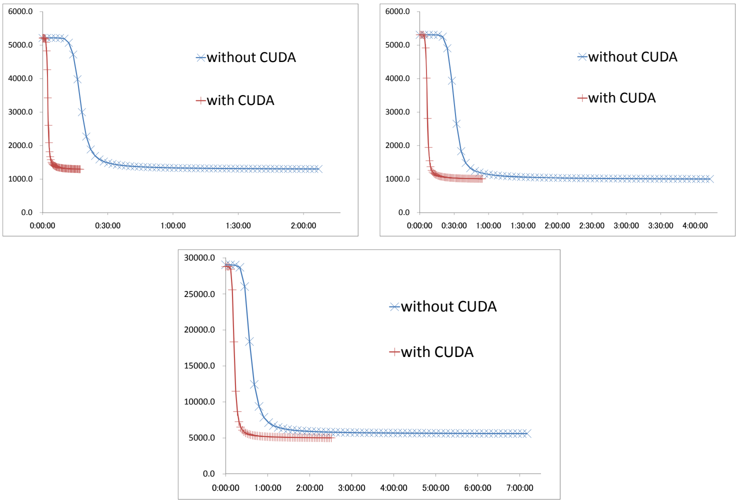


Fig. 5. Test data perplexity, averaged over 10 trials, versus execution time. Results for the newswire article set is presented in top row, where left and right panel correspond to $K = 64$ and $K = 128$, respectively. Results for the image dataset is given in bottom panel. CVB inferences with CUDA can achieve comparable (or even less) test perplexities in far shorter time for all settings.

because parallelized inferences may occasionally find better descent directions. Similar arguments can be found in a previous paper proposing parallelized collapsed Gibbs sampling for LDA [10]. For this image set, we provide a complete result, where we use full dataset for inference, at the Web site of an author (`http://www.cis.nagasaki-u.ac.jp/~masada/LDAimage/page1.html`).

Fig. 5 intuitively shows that we could obtain comparable perplexities in much shorter time with CUDA. Based on these results, we can conlude that our parallelization is in fact a promising computational improvement.

6 Conclusions

In this paper, we propose an acceleration of collapsed variational Bayesian inference for latent Dirichlet allocation with Nvidia CUDA compatible GPU devices. We could obtain inference results of quality comparable to original sequential version in much shorter time only by introducing negligible approximations.

We are now implementing our method on a cluster of PCs equipped with graphics cards. In the near future, this combination of our fine-grained parallelization and exisiting coarse-grained parallelization will achieve further acceleration of collapsed variational Bayesian inference for latent Dirichlet allocation.

References

1. NVIDIA CUDA, http://www.nvidia.com/cuda
2. MeCab, http://mecab.sourceforge.net/
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. In: Advances in Neural Information Processing Systems 14, pp. 601–608 (2001)
4. Chu, C.T., Kim, S.K., Lin, Y.A., Yu, Y., Bradski, G.R., Ng, A.Y., Olukotun, K.: Map-Reduce for Machine Learning on Multicore. In: Advances in Neural Information Processing Systems 19, pp. 306–313 (2006)
5. Steyvers, M., Smyth, P., Rosen-Zvi, M., Griffiths, T.L.: Probabilistic Author-Topic Models for Information Discovery. In: The 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 306–315 (2004)
6. Hofmann, T.: Probabilistic Latent Semantic Indexing. In: The 22nd International Conference on Research and Development in Information Retrieval, pp. 50–57 (1999)
7. Madsen, R.E., Kauchak, D., Elkan, C.: Modeling Word Burstiness Using the Dirichlet Distribution. In: The 22nd International Conference on Machine learning, pp. 545–552 (2005)
8. Nallapati, R., Cohen, W.W., Lafferty, J.D.: Parallelized Variational EM for Latent Dirichlet Allocation: An Experimental Evaluation of Speed and Scalability. In: ICDM Workshop on High Performance Data Mining, pp. 349–354 (2007)
9. Porteous, I., Newman, D., Ihler, A.T., Asuncion, A., Smyth, P., Welling, M.: Fast Collapsed Gibbs Sampling for Latent Dirichlet Allocation. In: The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 569–577 (2008)
10. Newman, D., Ascuncion, A., Smyth, P., Welling, M.: Distributed Inference for Latent Dirichlet Allocation. In: Advances in Neural Information Processing Systems 20, pp. 1081–1088 (2007)
11. Newman, D., Smyth, P., Steyvers, M.: Scalable Parallel Topic Models. Journal of Intelligence Community Research and Development (2006)
12. Teh, Y.W., Newman, D., Welling, M.: A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation. In: Advances in Neural Information Processing Systems 19, pp. 1378–1385 (2006)
13. Wang, X.G., Grimson, E.: Spatial Latent Dirchlet Allocation. In: Advances in Neural Information Processing Systems 20, pp. 1577–1584 (2008)
14. Xing, D.S., Girolami, M.: Employing Latent Dirichlet Allocation for Fraud Detection in Telecommunications. Pattern Recog. Lett. 28, 1727–1734 (2007)
15. Li, J., Wang, J.Z.: Automatic Linguistic Indexing of Pictures by a Statistical Modeling Approach. IEEE Trans. Pattern Anal. Mach. Intell. 25(9), 1075–1088 (2003)
16. Wang, J.Z., Li, J., Wiederhold, G.: SIMPLIcity: Semantics-sensitive Integrated Matching for Picture LIbraries. IEEE Trans. Pattern Anal. Mach. Intell. 23(9), 947–963 (2001)

Boosting Formal Concepts to Discover Classification Rules

Nida Meddouri and Mondher Maddouri

Research Unit on Programming, Algorithmics and Heuristics - URPAH,
Faculty of Science of Tunis - FST,
Tunis - El Manar University,
Campus Universitaire EL Manar, 1060, Tunis, Tunisia
nmeddouri@gmail.com,
mondher.maddouri@fst.rnu.tn

Abstract. Supervised classification is a spot/task of data mining which consists in building a classifier from a set of examples labeled by their class (learning step) and then predicting the class of new examples with a classifier (classification step). In supervised classification, several approaches were proposed such as: Induction of Decision Trees, and Formal Concept Analysis. The learning of formal concepts is based, generally, on the mathematical structure of Galois lattice (or concept lattice). The complexity of generation of Galois lattice, limits the application fields of these systems. In this paper, we present several methods of supervised classification based on Formal Concept Analysis. We present methods based on concept lattice or sub lattice. We also present the boosting of classifiers, an emerging technique of classification. Finally, we propose the boosting of formal concepts: a new adaptive approach to build only a part of the lattice including the best concepts. These concepts are used as classification rules. Experimental results are given to prove the interest of the proposed method.

Keywords: Formal Concept, Boosting, Classification rules, Machine Learning, Data mining.

1 Introduction

Formal Concept Analysis is a formalization of the philosophical notion of concept defined as a couple of extension and comprehension. The comprehension (called also intention) makes reference to the necessary and sufficient attributes which characterizes this concept. The extension is a set of examples which made it possible to find out the concept [8].

The classification approach based on *Formal Concept Analysis* is a symbolic approach allowing the extraction of correlations, reasons and rules according to the concepts discovered from data [4]. Supervised classification is a process made up of two steps. In the learning step, we organize the information extracted from a group of objects in the form of a lattice. In the classification step, we determine the class of new objects, based on the extracted concepts. Many learning

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 501–510, 2009.

methods based on *Formal Concept Analysis* were proposed, such as: *GRAND* [15], *LEGAL* [11], *GALOIS* [3], *RULEARNER* [16], *CIBLe* [14], *CLNN&CLNB* [19], *IPR* [12], *NAVIGALA* [10] and more recently *CITREC*[5].

In this last decade, a great number of researches in machine learning have been concerned with the boosting methods of classifiers that allow the improvement of the performances of a single learner by the techniques of vote [2]. The two principal reasons for this success are probably the simplicity of implementation and the recently theorems relative to the boundaries, the margins, or to the convergence [6] [17]. The boosting is known to improve the performance of any learning algorithm, supposed nevertheless unstable and discovering weak classifiers (weak learner). Unfortunately, systems based on *Formal Concept Analysis* encountered some problems such as an exponential complexity (in the worth case), a high error rate and over-fitting. Fortunately, boosting algorithms are known by improving the error rate of any single learner.

In this paper, we propose a new method based on *Formal Concept Analysis* and exploiting the advantages of boosting algorithms. This method uses the basic algorithm of multi class boosting: *AdaBoost.M2* [7]. Initially, the algorithm sets equal weights to the learning examples. It selects a part of the learning data and extracts the pertinent formal concept within the data sets. It uses the discovered concept to classify the learning data and updates the weights of the learning examples by decreasing those of the well classified ones and by increasing the weights of the others (the bad examples). After that, it repeats the resampling based on the new weights, in order to discard the well classified examples and to consider only the bad ones.

In section 2, we present the basics of *Formal Concept Analysis*. In section 3, we describe the basics of the Boosting and we present our method the *Boosting of Formal Concepts* (*BFC*). Then, we present experimental results in section 4 to prove the validity of our proposed approach.

2 Basics of Formal Concept Analysis

A formal context is a triplet $< \mathcal{O}, \mathcal{P}, \mathcal{R} >$, where $\mathcal{O} = \{o_1, o_2, ..., o_n\}$ is a finite set of elements called objects (instances, examples), $\mathcal{P} = \{p_1, p_2, ..., p_m\}$ a finite set of elements called properties (attributes) and $\mathcal{R}$ is a binary relation defined between $\mathcal{O}$ and $\mathcal{P}$. The notation $(o,p) \in \mathcal{R}$, or $\mathcal{R}(o,p) = 1$, which means that object o verifies property p in relation $\mathcal{R}$ [8]. The context is often represented by a cross-table or a binary-table as shown in Table 1 (taken from[13]).

Let $A \subseteq \mathcal{O}$ and $B \subseteq \mathcal{P}$ two finite sets. For both sets A and B, operators $\varphi(A)$ and $\delta(B)$ are defined as [8]:

- $\varphi(A) := \{ \ p \mid \forall \ o, \ o \in A$ and $(o,p) \in \mathcal{R} \ \}$;
- $\delta(B) := \{ \ o \mid \forall \ p, \ p \in B$ and $(o,p) \in \mathcal{R} \ \}$.

Operator φ defines the properties shared by all elements of A. Operator δ defines objects which share the same properties included in set B. Operators φ and δ define a Galois connection between sets A and B [8]. The closure operators

Table 1. Illustration of the formal context

$\mathcal{O}$-P	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	*CLASS* (Y)
o_1	1	1	1	1	1	1	1	0	***1***
o_2	1	1	1	1	1	1	0	1	***1***
o_3	1	1	1	1	1	0	1	1	***1***
o_4	1	1	1	1	0	1	0	0	***1***
o_5	1	1	0	1	1	0	1	0	***2***
o_6	1	1	1	0	1	0	0	1	***2***
o_7	1	0	1	0	0	1	0	0	***2***

are A"$= \varphi\circ\delta(A)$ and B"$=\delta\circ\varphi(B)$. Finally, the closed sets A and B are defined by $B=\delta\circ\varphi(B)$ and $A=\varphi\circ\delta(A)$.

A formal concept of the context $< \mathcal{O}, \mathcal{P}, \mathcal{R} >$ is a pair (A,B), where $A\subseteq\mathcal{O}$, $B\subseteq\mathcal{P}$ and $\varphi(A)=B$, $\delta(B)=A$. Sets A and B are called, respectively, the extent (domain) and intent (co-domain) of the formal concept.

From a formal context $< \mathcal{O}, \mathcal{P}, \mathcal{R} >$, we can extract all possible concepts. The set of all concepts may be organized as a complete lattice (called Galois lattice [8]), when we define the following partial order relation $\ll$ between two concepts, $(A_1,B_1)\ll(A_2,B_2)$ if and only if $(A_1\subseteq A_2)$ and $(\mathrm{B}_2\subseteq B_1)$. The concepts (A_1,B_1) and (A_2, B_2) are called nodes in the lattice. The Hass diagram is the chart of the Galois lattice [8].

3 Boosting of Formal Concepts

The idea of boosting is to build many classifiers (experts) who complement each other, in order to build a more powerful classifier . The boosting is an adaptive approach, which allows classifying correctly an instance, badly classified by an ordinary classifier. It allows decreasing exponentially the error rates of any classifier [7]. So, the boosting is a general method to convert a weak learner into an effective classifier. Two major interests for the combination of classifiers. First, a more reliable decision can be obtained by combining the opinion of several experts. Second, a complex problem can be decomposed into several sub problems which are easier to understand and to solve (divide and conquer).

The general idea of the algorithms which are based on the boosting is iterative. First, they select a subset of data from the training data set. Second, they build a classifier using the selected data. They evaluate the classifier on the training data set. And they start again T times. Two frequent questions tackle the approach based on boosting. How do we select the subsets? How do we combine the classifiers? In fact, the diversity of responses leads to a diversity of boosting algorithms. Among the most known ones, is the *Adaptive Boosting* (*AdaBoost*). Initially, *AdaBoost* assigns to the learning instances equal weights. It selects randomly a subset of the training examples. Then, it applies its learning algorithm on this subset to extract the resultant classifier. It calculates the error rate of the classifier over all the training set. So, if an instance is classified correctly by the classifier, *AdaBoost* increases its weight. Otherwise, it decreases its weight. It standardizes the weights of the entire instances [6] and it starts again all the procedure according to these conditions. The first algorithm is called *Adaboost.M1* [6],[7]. It repeats the previous process number of iterations fixed by the user in the beginning. If the error rate of one classifier becomes over 0.5, the current iteration is aborted and all the process is stopped. The second algorithm is called *AdaBoost.M2* [7]. It has the particularity to handle multi-class data and to operate whatever the error rate is.

3.1 The Proposed Method

Our approach is essentially based on *AdaBoost.M2* [7] described with more details in Algorithm 2. We execute, T times, the learning algorithm on various distributions of the examples. For each iteration, we do not obtain new examples but we are satisfied to perturb the distribution by modifying the weights of the learning examples.

Initially, the algorithm (the part based on *Adaboost.M2*: Algorithm 2) sets equal weights to the learning examples: $\mathcal{O}$. Then, our proposed learning algorithm (Algorithm 1) starts by selecting another set: O_t (from the previous selected set) through resampling the learning data by probabilistic drawing from $\mathcal{O}$:

$$O_t = \{(o_i, y_i) : i = \{1, ..., n'\}, n' \leq n\} \tag{1}$$

Our proposed learning algorithm mines O_t. It extracts the pertinent formal concept within the data sets O_t by selecting the attribute which minimise the function of Shannon entropy. If there are many attributes having the same entropy value, we select the one which has the higher support value. Once the attribute p^* is selected, we extract associated examples verifying it: $\delta(p^*)$. Then, we focus on the list of others attributes verifying all the extracted examples (using the closure operator $\delta \circ \varphi$ (p^*)).

So we construct our pertinent concept from the selected examples $(\delta(p^*), \delta \circ \varphi(p^*))$. A weak classifier is obtained by seeking the majority class associated to the extent of the pertinent concept. It induces a classification rule. The condition part of the rule is made up by the conjunction of the attributes included in the intent: $\delta \circ \varphi(p^*)$. The conclusion part of the rule is made up by the majority class. After that, it uses the discovered rule to classify the learning data set $\mathcal{O}$ and so our proposed learning algorithm of pertinent concept ends for this iteration.

So for each iteration, our algorithm generates a classifier h_t (a classification rule), in return to the algorithm of *AdaBoost.M2* (Algorithm 2). This classifier gives an estimated probability to the class y_i from the entry o_i. Three cases are presented:

- If $h_t(o_i,y_i) = 1$ and $h_t(o_i,y) = 0$, $\forall y \neq y_i$ then h_t has correctly predict the class of o_i.
- If $h_t(o_i,y_i) = 0$ and $h_t(o_i,y) = 1$, $\exists y \neq y_i$ then h_t has opposing predict class of o_i.

Algorithme 1. The learning algorithm of pertinent concept

```
INPUT: Sequence of $n$ examples $\mathcal{O} = \{(o_1,y_1),...,(o_n,y_n)\}$ with labels $y_i \in \mathcal{Y}$.
OUTPUT: The classifier rule.
BEGIN :
1. Select examples by a probabilistic drawing from $\mathcal{O}$ :
       $O_t = \{(o_i,y_i)$:i$\in\{1,..., n'\}, n' \leq n\}$.
2. From $O_t$, find the attribute which has less entropy value. Select the one
which has high support value if there are many attributes having the minimal
value of entropy: $p^*$.
3. Calculate the closure associated to this attribute in order to generate
the pertinent concept : $(\delta(p^*), \delta \circ \varphi(p^*))$.
4. Determine the majority class associated to $\delta(p^*)$ : $y^*$.
5. Induce the classification rule $h_t$: the conjunction of attributes from $\delta \circ \varphi(p^*)$
implied the membership to the same majority class $y^*$.
END.
```

- If $h_t(o_i,y_i) = h_t(o_i,y)$, $\forall y \neq y_i$ then the class of o_i is selected randomly between y and y_i.

From this interpretation, the pseudo-loss of the classifier h_t via the distribution W_t, is defined by:

$$\varepsilon_t = 0.5 \times \sum_{(o_i,y_i) \in O} W_t(o_i, y)(1 - h_t(o_i, y_i) + h_t(o_i, y)) \tag{2}$$

So we calculate the error

$$\beta_t = \varepsilon_t/(1 - \varepsilon_t) \tag{3}$$

and we update the weights of the examples according to β_t and Z_t (a normalization constant chosen so that $\sum_{i=1}^{n} W(o_i,y_i)$=1). Suppose that it updates the weights of the learning examples by decreasing those of the well classified ones and by increasing the weights of the others (the bad examples). After T iterations, we obtain the final classifier via:

$$h_{fin}(o_i) = \arg\max_{y \in \mathcal{Y}} \sum_{t=1}^{T} log(1/\beta_t) \times h_t(o_i, y_i) \tag{4}$$

In fact, the final classifier h_{fin} is a set of weak classifiers generated from the T iteration. Each iteration generates a weak classifier (a classification rule) or updates the weight of an existing weak classifier.

3.2 Illustrative Example

As an illustrative example, we start the first iteration on the initial formal context of Table 1. Algorithm 1 selects the examples by a probabilistic drawing from

Algorithme 2. AdaBoost M2 by Yoav Freund & Robert E.Schapire [7]

```
INPUT :
    Sequence of $n$ examples $\mathcal{O} = \{(o_1,y_1),...,(o_n,y_n)\}$ with labels $y_i \in \mathcal{Y}$.
    Weak learning algorithm (Algorithm 1).
    Integer $T$ specifying the number of iterations.
OUTPUT : the final hypothesis $h_{fin}$ classifier.
BEGIN :
1: INITIALIZE $W_t(o_i,y_i) = 1/|\mathcal{O}|$ for each $(o_i,y_i) \in \mathcal{O}$.
2: For $t$=1 to $T$
2.1: Call Weak Learn (Algorithm 1), providing it with mislabels distribution $W_t$.
2.2: Get back a hypothesis $h_t$: $O \times Y \longrightarrow [0,1]$.
2.3: Calculate the pseudo-loss of $h_t$ : $\varepsilon_t$.
2.4: Set $\beta_t = \varepsilon_t$ / (1- $\varepsilon_t$).
2.5: Update $W_t$ : $W_{t+1}(o_i,y_i) = {\beta_t}^{0.5\times(1+h_t(o_i,y_i)-h_t(o_i,y))} \times W_t(o_i,y)/Z_t$.
End For.
3: $h_{fin}(o_i)$ = arg max$_{y\in\mathcal{Y}} \sum_{t=1}^{T}$ log($1/\beta_t$)×h$_t(o_i,y_i)$.
END.
```

the initial context. It retains the examples: o_2, o_3, o_5. Then it compares the 8 attributes describing these examples and it chooses the attribute p_3 with a minimal entropy value of 0 and a maximal support value of 2 . The Galois function applied to this attribute provides the following formal concept ($\{o_2, o_3\}$,$\{p_1, p_2, p_3, p_4, p_5, p_8\}$). The formal concept with associated majority class, constructs a rule :

$$IF\ \ p_1 \wedge p_2 \wedge p_3 \wedge p_4 \wedge p_5 \wedge p_8\ \ THEN\ \ \boldsymbol{CLASS = 1} \tag{5}$$

With only 30 iterations, we obtain the 8 pertinent concepts. We associate each pertinent concept to the associated majority class and we obtain a list of the generated rules.

4 Comparative Study

In this section, we compare the proposed with existing ones based on *Formal Concept Analysis*: *IPR* [12] and *CITREC* [5]. To compare the presented approaches, we consider their complexities, their error rates and theirs numbers of generated concepts.

4.1 Comparison of Complexities

Concerning the complexities, the variable n means the number of examples and the variable m means the number of attributes. T is a variable which means the number of iterations. We examine the learning algorithm of pertinent concept and we estimate its complexity to *O(nlog(n) + nm)*. The complexity of probabilistic drawing from a set, is estimated to *O(nlog(n))* (due to the complexity of quick sort). The complexity of closure operator is *O(nm)*. Also, we examine the algorithm of *AdaBoost.M2* and we estimate its complexity. The complexity of the boosting iterations is $O(T)$. The complexity of applying our weak classifier is $O(nlog(n) + nm)$. The complexity of updating the weights of the examples is bounded by $O(n)$. To conclude, the complexity of the *BFC* method is estimated to $O(T(nlog(n)+ nm) + n) = O(nlog(n)+ nm)$.

To calculate the complexity of *CITREC* method, we re-examine the generation of reduced context and we estimate its complexity to $O(n^2 m)$. Also the complexity of generating lattice from the previous reduced context is estimated to $O(C\ 2^m)$ with C the number of classes. Then we examine the learning algorithm and we estimate its complexity to $O(n)$. To conclude, the complexity of *CITREC* method is estimated to $O(n^2 m\ +\ C2^m\ +\ 2^m n)$ and bounded by $O(2^m n)$.

The complexity of the *IPR* method is token from [12]. It's estimated to $O(n^2 m^2 (m+n))$.

Compared to the complexities of the other methods (See Table 2), we remark that the *BFC* method has the least theoretical complexity.

Table 2. Theoretical comparison of the methods: *CITREC*, *IPR* and *BFC*

Systems	*CITREC*	*IPR*	*BFC*
Kind of lattice	Sub-lattice	Cover	Sub-lattice
Algorithms	Godin	Maddouri2004	Closure Operator
Data	Binary	Binary	Binary
Number of classes	Multi-classes	Multi-classes	Multi-classes
Selection of concepts	No-inclus.+Supp.	Entro.	Entro.+Supp.
Combination of methods	Bay.-inf. or K-PPV	No	AdaBoost.M2
Knowledge learned	Rules	Rules	Rules
Classification	Vote	More weighted	Vote
Complexity	$O(2^m n)$	$O(n^2 m^2 (m+n))$	$O(nlog(n)+nm)$

4.2 Comparison of the Error Rates

To compare the methods based on the error rates, we used some known data sets from "*UCI Machine Learning Repository*" [1]. The chosen data sets were discretized with 2 discretional filters under *WEKA*[1]. The first is an instance filter that converts a range of numeric attributes into nominal attributes. Then a second filter is applied to convert all nominal attributes into binary attributes. Table 3 presents the characteristics of these data sets.

To calculate the error rates generated by each method applied to each data set, we use the *WEKA* software. *CITREC*, *IPR* and *BFC* are integrated in a modified version of *WEKA*. The experimentation is done by 10 cross-validation. We have to remember that the data sets are transformed beforehand to binary data (before treatment by these methods) using the *WEKA* filters.

As shown in Table 4, *BFC* has the specific ability to reduce the error rates compared to methods based on *FCA* (*CITREC* and *IPR*).

[1] Available at `http://www.cs.waikato.ac.nz/ml/Weka`

Table 3. Data sets specification

Data Set	Exemples	Attributes	Binary Attributes	Classes
Balance-Scale	625	4	4	3
Contact-Lenses	25	4	6	3
Diabets	768	8	8	2
Glass	214	9	9	6
Heart-statlog	270	13	13	2
Iris	150	4	4	3
Lymph	148	19	38	4
Segment	2310	19	19	7
Sonar	208	60	60	2
Weather.nominal	14	4	8	2

Table 4. Comparison of the error rates

Data Set	*CITREC*	*IPR*	*BFC*
Balance-Scale	40,47%	**35,83%**	46,72%
Contact-Lenses	55%	40%	**36,67%**
Diabetes	41,67%	-	**34,89%**
Glass	-	66,32%	**58,35%**
Heart-statlog	30,74%	-	**29,26%**
Iris	**20,67%**	50%	33,33%
Lymph	85,05%	-	**77,48%**
Segment	**33,9%**	-	84,89%
Sonar	53,38%	-	**36,02%**
Weather.nominal	35%	35%	**25%**

Table 5. Comparison of the numbers of concepts

Data Set	*CITREC*	*IPR*	*BFC*	Complete Lattice
Balance-Scale	7	4	**2**	16
Contact-Lenses	8	14	**5**	33
Diabets	4	-	**1**	256
Glass	-	49	**3**	232
Heart-statlog	4	82	**3**	3237
Iris	7	8	**3**	11
Lymph	16	180	**5**	6019
Segment	38	-	**5**	3037
Sonar	**4**	-	7	-
Weather.nominal	**4**	16	6	36

4.3 Comparaison of the Number of Concepts

It's very important to determine the number of concepts obtained by each system as shown in table 5. To determine the number of concepts in the concept lattice;

we use the *Galicia* [2] software [18] which construct the complet lattice of concepts [9].

As shown in table 5, the concept lattice contains a great number of concepts. *BFC* induces a small part of the lattice (On average 0.25%). It generates less than 10% of the concepts obtained by *IPR*. Also, it generates 40.22% of the concepts obtained by *CITREC*. *BFC* gives a lower number of concepts in less possible time compared to the other approaches.

5 Conclusion

In this paper, we have been interested in the classification approach based on *Formal Concept Analysis*. We have presented the methods based on concept lattice (i.e. *GRAND*), *IPR* based on cover of pertinent concept and *CITREC* based on concept lattice generated from a reduced formel context. Then, we have introduced the basics of boosting, especially the algorithm *AdaBoost.M2* and its advantages. Finally, we have given our proposed approach based on the *Boosting of Formal Concepts* that build adaptively a part of the concept lattice made up only by pertinent concepts. The method of *Boosting Formal Concepts* has the particularity to decide the number of iterations. So it can control the time of execution and give the best decision in T iterations. This new method is consolidated by a set of comparative studies. We show its polynomial complexity, making it faster than any other method based on *FCA*. Using known data sets, our method shows important benefits in terms of reduction in the number of concepts keeping a good precision.

Future work will focus on the quality of the classification rules. In fact, we plan to evaluate this method on other data sets. Accordingly, we plan to improve the error rate by acting on the voting methods and the measure of the best attribute. In practice, adopting simple approaches regularizes the learning. So, it is necessary to fix the number of iterations for the learning. T does not need to be known from the beginning. Many criteria exist to determine T [7]. In the future, we plan to study these criteria and to use them in our boosting method. In all cases, the outliers quickly destabilize the solution, when producing very important weights. For this fact, there exists heuristics to limit the effect of the examples or to detect / eliminate the outliers [7]. Also in the future, we plan to study these heuristics in order to improve the proposed method.

References

1. UCI machine learning repository, `http://www.ics.uci.edu/~mlearn/MLRepository.html`
2. Breiman, L.: Bagging predictors. J. Machine Learning 24, 123–140 (1996)
3. Carpineto, C., Romano, G.: Galois: An order-theoretic approach to conceptual clustering. In: International Conference on Machine Learning, pp. 33–40. Morgan Kaufmann Publications, Amherst (1993)

[2] Available at `http://www.iro.umontreal.ca/~galicia/`

4. Carpineto, C., Romano, G.: Concept Data Analysis: Theory and Applications. John Wiley and Sons, New Jersey (2004)
5. Douar, B., Latiri, C., Slimani, Y.: Approche hybride de classification supervisèe à base de treillis de galois: application à la reconnaissance de visages. In: Conférance Extraction et Gestion des Connaissances, pp. 309–320. RNTI, Sophia, Sophia-Antipolis (2008)
6. Freund, Y.: Boosting a weak learning algorithm by majority. J. Information and Computation 121, 256–285 (1995)
7. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: International Conference on Machine Learning, pp. 148–156. Morgan Kaufmann Publications, Bari (1996)
8. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, New York (1997)
9. Godin, R., Missaoui, R., Alaoui, H.: Incremental concept formation algorithms based on galois (concept) lattices. J. Computational Intelligence 11, 246–267 (1995)
10. Guillas, S., Bertet, K., Ogier, J.-M.: Reconnaissance de symboles bruités à l aide d un treillis de galois. In: Colloque International Francophone sur l Ecrit et le Document, pp. 85–90. IDIAP, Fribourg (2006b)
11. Liquiere, M., Mephu Nguifo, E.: Legal: learning with galois lattice. In: Bruneau, L. (ed.) Journées Franaises sur l'Apprentissage, Lannion, pp. 93–113 (1990)
12. Maddouri, M.: Towards a machine learning approach based on incremental concept formation. J. Intelligent Data Analysis 8, 267–280 (2004)
13. Nguifo, E.M., Njiwoua, P.: Treillis de concepts et classification supervisèe. J. Technique et Science Informatiques 24, 449–488 (2005)
14. Njwoua, P., Mephu Nguifo, E.: Améliorer l'apprentissage à partir d'instances grâce à l'induction de concepts: le système cible. Revue d'Intelligence Artifcielle 13, 413–440 (1999)
15. Oosthuizen, D.: The use of a Lattice in Knowledge Processing. Thesis university, University of Strathclyde, Glasgow (1988)
16. Sahami, M.: Learning classification rules using lattices. In: European Conference on Machine Learning, pp. 343–346. Springer, Heraclion (1995)
17. Sebban, M., Suchier, M.: On boosting improvement: Error reduction and convergence speed-up. In: European Conference on Machine Learning, pp. 349–360. Springer, Heidelberg (2003)
18. Valtchev, P., Grosser, D., Roume, C., Hacene, M.R.: Galicia: an open platform for lattices. In: International Conference on Conceptual Structures, pp. 241–254. Shaker Verlag, Dresden (2003)
19. Xie, Z., Hsu, W., Liu, Z., Lee, M.-L.: Concept lattice based composite classifiers for high predictability. J. Experimental and Theoretical Artificial Intelligence 14, 143–156 (2002)

Supervised Classification Algorithms Applied to the Detection of Atmospheric Pollution

V. Delcroix, F. Delmotte, D. Gacquer, and S. Piechowiak

LAMIH - UVHC, UMR CNRS, University of Valenciennes, Le Mont Houy,
59313, Valenciennes Cedex 9, France
{veronique.delcroix,francois.delmotte,david.gacquer,
sylvain.piechowiak}@univ-valenciennes.fr

Abstract. The management of atmospheric polluting reject is based on the ability to measure this pollution. This paper deals with the case where no local sensor can be used, inducing the use of video to detect and evaluate the atmospheric pollution coming from large industrial facilities. This paper presents a comparison of different classifiers used in a monitoring system of polluting smokes detected by cameras. The data used in this work are stemming from a system of video analysis and signal processing.

The database also includes the pollution level of plumes of smoke defined by an expert. Several Machine Learning techniques are tested and compared. The experimental results are obtained from a real world database of polluting rejects. The parameters of each type of classifier are split in three categories: learned parameters, parameters determined by a first step of the experimentation, and parameters set by the programmer. The comparison of the results of the best classifier of each type indicates that all of them provide good results.

Keywords: Machine Learning, Supervised classification, Air Pollution.

1 Introduction

The consideration in environment preservation and the reduction of greenhouse effect gas emission has been considerably growing during the last decade. This consideration has led to the establishment of international environment law and more particularly the "*Polluter Pays Principle*" where the polluting party pays for the damage done to natural environment. It is also regarded as a regional custom, because of the strong support local communities have to bring regarding the installation of recycling facilities and pollution and waste control equipment. This growing concern of environment control has drawn the interest of researchers in a wide range of applications. This is the context of the present paper, which focuses on the control of air pollution and addresses this control as a supervised classification application. The goal is to determine automatically the impact and the gravity of hazardous smokes rejected by factories. Most of the studies concerning pollution detection systems are based on chemical analysis, using either chemical measurements or laser detection [1], and little attention has been given to automatic systems for monitoring industrial facilities and classifying aerial pollution. Unlike usual pollution control systems, our

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 511–522, 2009.

goal here is to use visual information from a camera to detect and to predict the potential danger of polluting rejects.

The choice of using a camera instead of local sensors is justified by the structure of the factories (like steel industries) where the system is used and by the difficulties in using sensors. In addition it is necessary that the system must be online.

This paper addresses the classification of hazardous smoke using image analysis and Machine Learning algorithms. It is organized as follows: in the next section, the monitoring system designed for automatic detection of polluting smokes is detailed. Then, the following section presents the 6 methods of supervised classification used for this problem and the obtained results. The last section is dedicated to a discussion and the conclusion.

2 Experimental Material

The ALOATEC Company designs mainly monitoring and quality control solutions for its industrial partners. This company has developed the DETECT system, a product designed for the visual detection of polluting smokes around large industrial complexes.

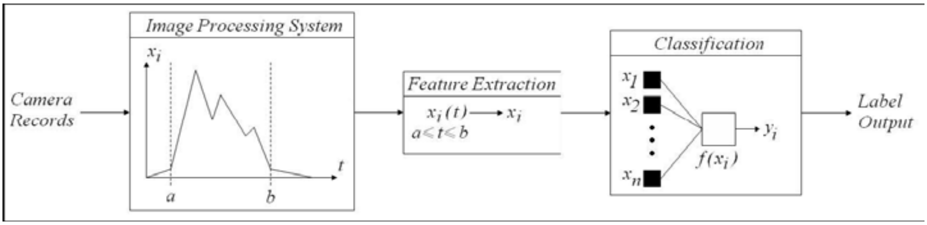


Fig. 1. General architecture of the DETECT system

The development of this system is motivated by the need to have a detailed and quantified summary of all hazardous events that occur in the region. The original classification program developed by ALOATEC is a user defined expert system. This solution requires one or more human experts familiar with the industrial complex where the DETECT system is being used, to adjust the camera settings, but also to define the different rules of the classification program. This task can be especially complex since pollution can take various forms according to the activities of a given industrial complex. This implementation implies heavy constraints concerning installation and evolution of the system. Besides, the efficiency of expert systems mainly relies on the possibility to formalize the problem into an accurate and finite set of rules. Due to the complexity of the process, the first implementation of the system achieved an important number of misclassified objects. Moreover, it was not possible to adapt the system to different data sets from other problems. Another lack of the first monitoring system of polluting smoke is the high number of misdetection and over detection of severe pollution. To solve the limitation of the first system and improve its classification accuracy, the authors replace the original classification program with machine learning algorithms adapted to the pollution detection problem.

Since the first implementation of the system has been used for a certain period over different industrial sites, a sufficient database of samples is available as training material for the work exposed in this paper.

2.1 Information Extraction from Pictures

The camera monitors the activity of a given complex in a continuous manner, that is, image processing continuously translates visual information from the scene into numerical variations of different signals. The camera is placed outside the complex, so that all the areas where hazardous rejections may occur fall into the range of the picture. A typical example of plume is the left of Fig. 2. To protect the camera from weather conditions and to limit the introduction of noise in the picture, it is placed into a box behind a glass panel, so that its objective remains clear.

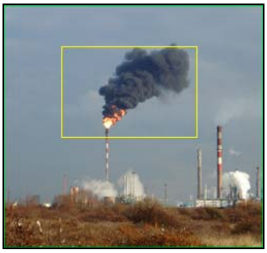


Fig. 2. Typical scene detected by camera

A detection algorithm determines time windows corresponding to potentially hazardous trail rejections. (cf. time interval [a,b] in Fig. 1.). This is done thanks to a frequency analyze on a small windows of the video image. While the smoke continues, several signals are recorded into a temporary database. As soon as the end of the plume is detected, the features are extracted from these records by using function on the time interval. The functions such as min, max, integral, mean, or delta between max and min depend on the physical signification of the signal. Each detected trail of smoke constitutes an event and is recorded with a frequency of one picture every five seconds. The features of the plume are stored into a permanent database together with the corresponding pictures.

In order to extract the features of the plume from the filmed sequence, several zones are cut out of the image: a positioning zone, a dynamic zone and four static zones. The centre of the emission is determined thanks to the positioning zone. The dynamic zone is automatically positioned on the plume as soon as one is detected. The static zones are four little areas of the image without any intersection with the plume. These zones allow to measure the difference between pixel marked as pollution and those marked as no pollution. The Table 1 gives what are the signals extracted from the filmed sequence corresponding to a plume of smoke.

The Machine Learning algorithms do not directly exploit these features. The temporal data are transformed in order to obtain quantitative variables that characterize the corresponding plume. One of the following transformations is applied

Table 1. List of the signals corresponding to a plume of smoke

Signal Number	Name	Explanation
1	luminosity	Luminosity in a upper corner of the image, without any pollution used to detect the nightfall
2	surface	Percentage of pixels of the dynamic zone labelled as pollution is computed
3…6	Luminosity of static zones	luminosity of the four little static areas defined on the filmed image as a reference
7	luminosity of dynamic zone	Luminosity of the most representative point labelled as pollution in the dynamic zone
8…15	colour of the static zones	Coordinates on the XY chromatic diagram of the colour filmed in the four static zones
16,17	colour of the dynamic zone	Coordinates on the XY chromatic diagram of the colour of the most representative point labelled as pollution in the dynamic zone
18…21	low, medium, high and very high density	These values are computed as follows: each point of the dynamic zone labelled as pollution is considered as the center of a line window of 21 pixels: 10 at left and 10 at right. The number of pixels of this window labelled as pollution defines the weight of the central point. The signals 18 to 21 are the sum of these weights that are respectively higher than 1, 8 13 and 17 / 20
22	Max Emission	This signal reproduces the evolution of the importance of the plume during the time window

on a signal or a combination of these signals: integral, mean, maximum, minimum or delta between max and min. Most of the above signals have been aggregated (by product for example) in order to reduce the number of variables. This step has been achieved by hand without significant loss of accuracy on the quality of the classifier on several individual tests. This process explains, as mentioned in the first section of this paper, why the activity of the various factories is monitored as a succession of discrete events and not like a continuous process. Also, unlike conventional pollution detection systems, the control is performed through visual information from the trail and not from its chemical analysis.

2.2 The Database

The database consists in 2900 events recorded during several months of activity on the same industrial factory. The database contains no missing value. A plume of smoke represents an event, or an object to be classified. Each object is characterized by 13 numerical features, including the class label. The 12 features are computed from the 22 signals described above and correspond to color, density, surface or contrast measurements determined by the image-processing component of the system. The details of the aggregation and transformation of the temporal signal into a scalar cannot be explained here due to confidentiality reasons. The class label is the severity level of the plume defined by a human expert from the domain. The level ranges from 0 to 3 and zero level correspond to a water vapor plume, whereas a level three correspond to the high pollution. The different classes are not equally represented in the database, since there is a large amount of degree 0 pollutions compared to the other classes. The class distribution is given in Tab. 2. The low ratio of critical cases makes more difficult the learning step of the classifiers despite the high number of learning cases.

Table 2. Class distributions in the experimental database

	Class 0	**Class 1**	**Class 2**	**Class 3**
Samples	2085	421	143	235
Ratio	72 %	15 %	5 %	8 %

3 Classification Problem

The classification problem [2] can be expressed as following: let us consider a set of training samples that consist in association (input vector, output value). The classification problem consists in finding a function *f* that maps each input vector into the corresponding class label. The learned function is expected not only to minimize the number of misclassified samples over the training data, but also to achieve high generalization properties when facing unknown samples [3].

In this paper we have selected and tried 6 common algorithms of classification: k-nearest neighbors (KNN) [4], decision tree (DT) [5], naive Bayesian network (NBN) [6], Multilayer Perceptron (MLP) [7], Support Vector Machine (SVM) [8] and Takagi-Sugeno Fuzzy model (TSF) [9].

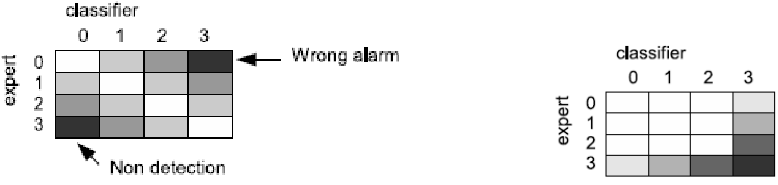
3.1 Setting the Parameters of the Classifiers

The task of defining and comparing classifiers is generally divided in two distinct steps: learning some parameters of the classifier from the data, testing the classifier in order to compare its performance with others. In this paper, an additional step has been added to determine the value of some of the parameters of the classifier that are best adapted to the particular data concerning the plumes of smoke. During the first step, parameter selection is performed for each classifier using 10-fold cross validation [10]. This method consist in cutting the initial set of data in 10 equal parts, and then running 10 times the learning of the classifier and its testing. Each time, the learning is done using 9 parts and the test is done on the tenth part. On each run, the part used for test is changed. The 10 fold cross validation is repeated for different values of each parameter in an interval. The value of each parameter that gives the best results, both for efficiency and for accuracy is used during the second step which is the comparison of the classifiers. During the second step, the whole data is used with the parameters determined previously for each algorithm. Average accuracy and efficiency are computed using holdout validation: the database is divided in two parts: 60% for learning and 40% for testing. This is averaged on 50 runs.

The first step has been achieved for the classifiers KNN, decision tree (DT) , MLP and SVM in order to determine the best value of one or more parameter. These parameters are the number k for KNN, the minimal number of objects on a node of a decision tree needed to split the node, the number of hidden units and of hidden layers for MLP, and the cost of constraint violation and the kernel parameter sigma (for Gaussian kernel) for SVM. The results of this step are detailed in section 3.

3.2 Measuring the Quality of a Classifier: Accuracy and Efficiency

The usual measure of the performance of a classifier is the accuracy, defined as the proportion of well classified objects. The confusion matrix (cf. Fig. 3a) displays in the cell *N(i,j)* the proportion of events classified in the class *i* by the expert and in the class *j* by the classifier.



(a) (b)

Fig. 3. (a) “Confusion” matrix. (b) Relative importance of misclassified object in the efficiency computation.

Some events misclassified by the system are more important in this real world situation. False alarms (actual degree 0 event labeled as degree 3 by the system) and missed detections (actual degree 3 labeled as degree 0 by the system) have a direct influence for industrial applications. On the contrary, if the system classifies an actual degree 1 event as a degree 0, the importance of the error is reduced. To consider classification accuracy in a practical view for industrial applications, an efficiency criterion is defined additionally to the ratio of correctly classified samples. This criterion reflects the number of correctly classified false alarms and missed detections:

$$accuracy = \frac{\sum_{i=0}^{3} N(i,i)}{\sum_{i=0}^{3}\sum_{j=0}^{3} N(i,j)} \qquad efficiency = \frac{0.3\times(N(1,3)+N(3,1)+0.8\times(N(2,3)+N(3,2)+N(3,3)}{\sum_{i=0}^{3} N(i,3)+\sum_{j=0}^{2} N(3,j)}$$

where *N(i,j)* is the number of objects whose actual class is *i* that were labeled as class *j* by the classifier. This criterion has been defined by the industrial expert. The relative importance of misclassified objects in the efficiency criterion is drawn on Fig. 3b.

Each of the classifiers compared in this paper are presented in this section. Discussions about parameter selection are also provided, regarding the classification techniques that require parameters adjustment. The values of the parameters that are directly set by the authors are given, as well as the values of the parameter that are determined by 10 fold cross. These values are used in the final step of learning of the third categories of parameters.

Experiments are run on a Pentium IV CPU with 1 Go of memory, using Matlab environment. The following Matlab toolbox have been used: the BNT-SLP package [11] for Bayesian network, the OSU-SVM Matlab toolbox for SVM, the ***nnet*** package for neural network, the ***stat*** package for decision tree and fuzzy model.

4 Comparison of the Classifiers

This section compares the results of 6 types of classifiers described in the previous section on a real world database. Each classifier has been built in order to fit as well as possible to this particular problem. The tables 3,4 and 5 summarizes the parameters of these classifiers.

As explained in the previous section, most of the algorithms exposed in this paper require parameter selection prior to the evaluation of classification accuracy. The parameters of the classifiers that have been directly set by the authors are presented in Table 3. The parameters concerned by the learning step are listed in Table 4. Those determined by selection, using 5 runs of a 10-fold-cross validation [10], are summarized in Table 5. The value of these parameters that maximized the performances of the classifier is fixed for the rest of the experiment and summarized in Table 5

4.1 Robustness of Classifiers towards the Parameters

If we consider that each value of the parameters listed in Table 3, 4 and 5 provides a different classifier, the number of possibilities offered to the user is very large, making more difficult the choice of a classifier. The objective of the user is to select easily a classifier that will “correctly” classify his data. Facing this problem, the analysis of the previous curves shows that most of the parameters that we considered have a moderate influence on the performances of the classifiers.

Concerning MLP, the number of hidden units has no significant influence on the performances over 3 hidden units with one hidden layer and over twice 3 with two hidden layers. However, a too large number of hidden units loads down unnecessarily the classifier and increases the learning time. Speaking of decision trees, the minimum number of observations needed for a node to be split is not a decisive parameter in the interval of value that we considered. The classifier KNN is slightly more sensible to the choice of the number k of neighbors. In the industrial case that we considered, the criterion of efficiency leads to a value of k between 3 and 7, whereas the usual criterion of efficiency leads to a choice with $k > 10$. For these three types of classifiers, MLP, KNN and decision tree, the difference of performances does not exceeds 3% for a large range of values of the parameters.

An exception must be underlined for the SVM with Gaussian kernel function, for which a bad value of the parameter σ can drastically decrease the performances of the classifier. It has to be mentioned that the correct interval of values of this parameter depends on the problem. To a slightest extent, the value of the cost of constraint violation is also important since the efficiency can increase up to 10% in the problem we address. It seems that for a good value of σ, the cost has to be set as high as possible, depending on what is considered as reasonable in term of training time of the SVM.

Table 3. Performances of the classifiers depending on the size of the learning set

Classifier	User defined Parameters
KNN	Metric Distance : Euclidien distance
DT	Learning algorithm : CART [5] Minimal number of observations for a node to be split : 10 Pruning : compute the full tree and the optimal sequence of pruned subtrees Criterion for choosing a split : Gini's diversity index
NBN	Graph Structure : star graph Discretization algorithm : hist-ic (optimal histogram based on IC information criterion [11]) Decision rule *: class = argmaxCP(C j X)*
MLP	Nb of hidden layers : 1 Nb of output units : 4 Decision rule : *class* = *argmaxCfC*(*x*) where *fC*(*x*) is the output of unit associated with class *C* Learning algorithm : Levenberg-Marquardt algorithm
SVM	Kernel function Gaussian Kernel function Multi-class implementation *: One Versus One model*
TSF	Membership functions : Gaussian Nb of rules : 5 Learning algorithm settings : an alternative of Levenberg-Marquardt algorithm

For each of the 6 classifiers, the learning step and the computation of the performances has been computed for different size of the training set, but always on 50 runs. For that purpose we run *n* times a *k*-fold cross validation with the four values of the couple *(n,k)* : (25,2), (10,5), (5,10), (2,25). For each run, *(k-1)/k %* of the samples are randomly chosen to train the classifier, and the remaining *1/k %* is used to compute accuracy and efficiency. Since the database includes 2900 events, the 2-fold cross validation corresponds to a learning set of 1450 lines, whereas the 25-fold cross validation leads to a learning set of size 2784. For the particular case of MLP, only two thirds of the training samples are used to learn the target function, the remaining third being used as a validation set to perform early-stopping.

Table 4. Summary of learned parameters

classifiers	Learned parameters
Decision Tree	Tree structure
Naive Bayes	Probability distributions
MLP	Weights *wij*
SVM	Support vectors
TS Fuzzy model	Premises: centers and bases of the membership functions Consequents: coefficients *aij* and bias *bi* of the linear function

Table 5. Summary of parameter determined by cross validation for each classifier

classifiers	CV determined Parameters	values
KNN	Number k of neighbors	9
Decision tree	Minimal number of observations for a node to be split	30
MLP	Number of hidden units	4
SVM	Cost of constraint violation C	10^4
	Kernel parameter σ	10^{-9}

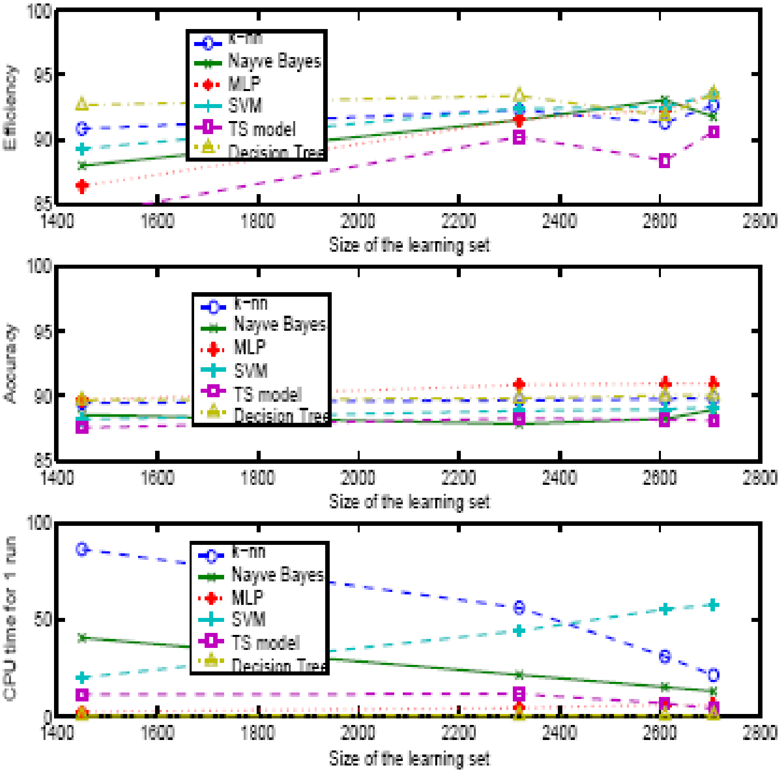


Fig. 4. Influence the size of the learning set over performances of six basic classifiers

Performances of the different classifiers with their optimal parameter settings (Tab. 5) are presented in Fig. 4. The CPU time is the total time needed for the 50 runs, for the learning step and the classification step. The original DETECT classification system based on an expert system performs much less efficiently than any other of the presented classifier since its accuracy equals 70.9% and its efficiency equals 48.2%. The different algorithms used in replacement of the expert system clearly outperform the original classification application, but more importantly, supervised learning allows the design of classification procedures automatically from the data, with good overall results, which considerably improves portability of the system and reduces the necessity of human intervention.

4.2 Robustness of the Classifiers Facing the Case of Under-Represented Classes

The cases of the database we have used are representative of a number of real cases where the class of interest is the least represented. Fortunately, hazardous puff of smoke are much less frequent that simple water vapour smoke. And it is the same for cancer studies, water pollution or default detection.

We consider that the efficiency used in this paper is a criterion that allows taking in account the robustness of the classifier facing under represented classes since it gives more importance to misclassified examples on these classes. In our case, and provided that the learning 24 set is large enough, the NBN seems to be a good choice of classifier. About this, it is necessary to remind that NBN has a very simple structure with the central class node directly linked with all the node of parameters. Lots of other structures are possible for the graph of the Bayesian network.

When the criterion of choice of the classifier is the accuracy, as it is frequently the case, the best choice seems to be MLP. It is however interesting to remark that the best three classifiers are also the best three classifiers in term of efficiency in the case of a smaller learning set.

In the case where a class of interest is weakly represented, the global error rate of the classifier may be an inadequate criterion to set the value of a parameter in order to select the classifier. This is what is observed for the SVM where the parameter σ for the Gaussian kernel has to be set with attention. For a bad value of sigma, the accuracy keeps a value over 70%, which may not draw the attention of the unadvertised user, whereas the efficiency is null. The same case is observed with a MLP with two hidden layer, with insufficient number of hidden unit: the efficiency fall down under 30% whereas the accuracy is over 80%.

The original expert system offered an accuracy of 70.9 and an efficiency of 48.0. The average performances of all the classifiers we test clearly outperform the original. However, none of these classifiers is clearly better than the others for all the following criteria of choice: accuracy or efficiency and the size of the available learning set. The neural networks perform well both in term of accuracy and efficiency, the number of hidden layer and hidden units are note determinant for the quality of this classifier. The same conclusion can be drawn for decision trees with accuracy 2% less good. The KNNs perform well but possibly need a long time to achieve the classification since they have no prior learning step. NBN are an interesting solution in case the learning set is large enough. The SVMs would not be the first choice, mainly because of the difficulty to set correctly the parameter of the Gaussian kernel. The performance of SVMs and the TSFs, although comparable to the other's are slightly less good.

5 Conclusions

This paper presents a comparative study of several algorithms adapted to supervised classification of camera-detected polluting smokes. 6 Machine Learning algorithms are used and the benefits offered by each one are compared: all of them outperform the original classification device of the DETECT system (in terms of efficiency and correct classification).

The results exposed show that it is possible to learn a good classification procedure directly from the data for our real world application without requiring several human experts to define settings on each industrial factory where the system is used. This is a crucial point, since one of the goals is to design a system which can easily be used to monitor different industrial activities without requiring expert human intervention.

As also described in [12], comparing the different classifiers is a difficult task, since a given algorithm can outperform the others on a particular problem, and provided that its parameters are optimized. The confidence in the data is also very important. Some classifiers are more robust regarding noisy training sets, but even the most efficient algorithm will provide poor results if the information available is not relevant enough to perform classification. Some classifiers, like ANNs, are also unstable, which means that average error can vary from one or several percents between different series of tests.

Since there is no major difference between the algorithms, in terms of misclassification error, the constraints of the application have a greater influence concerning the choice of the optimal classifier. Since classification of polluting trails is performed online, the system has to give a fast response if two events occur successively. For this reason, a solution based on KNN is not advised, since classification of new objects can take several minutes if the number of training samples is important. ANNs or SVMs are more suitable for real time application because they are trained offline, and once classifier parameters are well estimated, time required to classify a single object is small enough to proceed with successive events.

Regarding future research work, we plan to improve classification accuracy by designing multi-classifier systems, to combine the advantages of different algorithms and exploit diversity in classifier sets.

Acknowledgments. This work was supported in part by the French Ministry of National Education and Research, under a MRT Grant, and the French Agency for Environment and Energy Mastery (ADEME). The authors would also like to thank P. Bourrier (ALOATEC Company) for their support concerning the research project exposed in this paper, and for having provided the data required for the experiments.

References

1. Bakos, G., Tsagas, N.: A laser system for pollution detection. Journal of The Franklin Institute 332, 211–218 (1995)
2. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, Hoboken (2001)
3. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)

4. Shakhnarovich, G., Darrell, T., Indyk, P.: Nearest-Neighbor Methods in Learning and Vision: Theory and Practice. In: Neural Information Processing. MIT Press, Cambridge (2006)
5. Quinlan, R.J.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
6. Pearl, J.: Causality: Models, Reasoning, and Inference. Cambridge University Press, Cambridge (2000)
7. Ripley, S.: Pattern recognition and neural networks (1996)
8. Vapnik, V.: Statistical Learning Theory. Wiley-Interscience, Hoboken (1998)
9. Zadeh, L.A.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. on Systems, Man, and Cybernetics SMC-3, 28–44 (1973)
10. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: IJCAI, pp. 1137–1145 (1995)
11. Leray, P., François, O.: Bnt structure learning package: documentation and experiments. Technical Report FRE CNRS 2645, Laboratoire PSI - INSA Rouen, Rouen, France (2004)
12. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience, Hoboken (2004)

A Hierarchy of Twofold Resource Allocation Automata Supporting Optimal Sampling

Ole-Christoffer Granmo[1] and B. John Oommen[2,⋆,⋆⋆]

[1] Dept. of ICT, University of Agder, Grimstad, Norway
[2] School of Computer Science, Carleton University, Ottawa, Canada

Abstract. We consider the problem of allocating limited sampling resources in a "real-time" manner with the purpose of estimating multiple binomial proportions. More specifically, the user is presented with 'n' sets of data points, $S_1, S_2, \ldots, S_n$, where the set S_i has N_i points drawn from two classes $\{\omega_1, \omega_2\}$. A random sample in set S_i belongs to ω_1 with probability u_i and to ω_2 with probability $1 - u_i$, with $\{u_i\}$. $i = 1, 2, \ldots n$, being the quantities to be learnt. The problem is both interesting and non-trivial because while both n and each N_i are large, the number of samples that can be drawn is bounded by a constant, c. We solve the problem by first modelling it as a *Stochastic Non-linear Fractional Knapsack Problem*. We then present a completely new on-line Learning Automata (LA) system, namely, the *Hierarchy of Twofold Resource Allocation Automata* (H-TRAA), whose primitive component is a *Twofold Resource Allocation Automaton* (TRAA), both of which are asymptotically optimal. Furthermore, we demonstrate empirically that the H-TRAA provides *orders of magnitude* faster convergence compared to the LAKG which represents the state-of-the-art. Finally, in contrast to the LAKG, the H-TRAA scales sub-linearly. Based on these results, we believe that the H-TRAA has also tremendous potential to handle demanding real-world applications.

Keywords: Estimation, Learning Automata, Stochastic Optimization.

1 Introduction

In this paper we consider the fascinating problem of learning distributions when the number of "Classes" is large, and the number of elements (i.e., data points) per class is also large, but when the resources available are limited. In particular, we concentrate on methods which are forced to resort to sampling techniques because of the cardinalities of the set of classes and of the set of data points per class.

The theory of estimation has been studied for hundreds of years. Besides, the learning (training) phase of a statistical pattern recognition system is, indeed,

⋆ *Chancellor's Professor*; *Fellow : IEEE* and *Fellow : IAPR*. The Author also holds an *Adjunct Professorship* with the Dept. of ICT, University of Agder, Norway.

⋆⋆ The second author was partially supported by NSERC, the Natural Sciences and Engineering Research Council of Canada.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 523–534, 2009.

based on estimation theory. Estimation methods generally fall into various categories, including the Maximum Likelihood Estimates (MLE) and the Bayesian family of estimates which are well-known for having good computational and statistical properties.

The problem studied can be stated as follows: We consider the case when the user is presented with 'n' sets of data points, $S_1, S_2, \ldots, S_n$.The set S_i has N_i points drawn from two classes $\{\omega_1, \omega_2\}$. A random sample in set S_i belongs to ω_1 with probability u_i and it belongs to ω_2 with probability $1 - u_i$. The problem we study involves estimating $\{u_i\}$ for $i = 1, 2, \ldots n$. However, to render the problem both meaningful and non-trivial, we assume that we are allowed to sample *a maximum of only* c points from the sets, $\{S_i\}$. Thus we have to determine how samples must be drawn (with replacement) from each set so that we can obtain both accurate and efficient estimates of $\{u_i\}$. The purpose is to make the estimates as accurate as possible, and so, we will here pursue the goal of minimizing the variance of the proportion estimates when the total number of samples available for estimating the proportions is restricted to c.

We shall utilize the model of the Stochastic Non-linear Fractional Equality Knapsack (NEFK) problem to model the present problem, and once such a formalization has been established, we shall allude to the LA solution of the NEFK to solve sampling problems.

In order to appreciate the qualities of the Stochastic NEFK Problem, it is beneficial to view the problem in light of the classical *linear* Fractional Knapsack (FK) Problem. Indeed, the Stochastic NEFK Problem generalizes the latter problem in two significant ways. Both of the problems are *briefly* defined below.

The Linear Fractional Knapsack (FK) Problem: The linear FK problem is a classical continuous optimization problem which also has applications within the field of resource allocation. The problem involves n materials of different value v_i per unit volume, $1 \leq i \leq n$, where each material is available in a certain amount $x_i \leq b_i$. Let $f_i(x_i)$ denote the value of the amount x_i of material i, i.e., $f_i(x_i) = v_i x_i$. The problem is to fill a knapsack of fixed volume c with the material mix $\boldsymbol{x} = [x_1, \ldots, x_n]$ of maximal value $\sum_1^n f_i(x_i)$ [1].

The Nonlinear Equality FK (NEFK) Problem: One important extension of the above classical problem is the *Nonlinear Equality* FK problem with a separable and concave objective function. The problem can be stated as follows [2]:

$$\begin{array}{l} \text{maximize } f(\boldsymbol{x}) = \sum_1^n f_i(x_i) \\ \text{subject to } \sum_1^n x_i = c \text{ and } \forall i \in \{1, \ldots, n\}, x_i \geq 0. \end{array}$$

Note that since the objective function is considered to be concave, the value function $f_i(x_i)$ of each material is also concave. This means that the derivatives of the material value functions $f_i(x_i)$ with respect to x_i, (hereafter denoted f_i'), are non-increasing.

The Stochastic NEFK Problem: In this paper we generalize the above NEFK problem. First of all, we let the material value per unit volume for any x_i be a *probability* function $p_i(x_i)$. Furthermore, we consider the distribution of $p_i(x_i)$

to be *unknown*. That is, each time an amount x_i of material i is placed in the knapsack, we are only allowed to observe an instantiation of $p_i(x_i)$ at x_i, and not $p_i(x_i)$ itself. As an additional complication, $p_i(x_i)$ is nonlinear in the sense that it decreases monotonically with x_i, i.e., $x_{i_1} \leq x_{i_2} \Leftrightarrow p_i(x_{i_1}) \geq p_i(x_{i_2})$. Given this stochastic environment, we intend to devise an on-line incremental scheme that learns the mix of materials of maximal *expected* value, through a series of informed guesses.

Stochastic Knapsack Problems — State-of-the-Art: The first reported generic treatment of the stochastic NEFK problem itself can be found in [3]. Various instantiations of the problem have, however, appeared sporadically, particularly within the web monitoring domain. In these latter instantiations, the unknown parameters are *estimated* by means of a tracking phase where web pages are polled mainly for estimation purposes [4]. One major disadvantage of such an approach is that the parameter estimation phase significantly delays the implementation of an optimal solution. This disadvantage is further aggravated in *dynamic* environments where the optimal solution changes over time, introducing the need for parameter re-estimation [3].

Learning Automata (LA): In contrast to the above approaches, we base our work on the principles of LA [5]. LA have been used to model biological systems [6], and have attracted considerable interest in the last decade because they can learn the optimal actions when operating in (or interacting with) unknown stochastic environments. Furthermore, they combine rapid and accurate convergence with low computational complexity.

The novel Learning Automata Knapsack Game (LAKG) scheme that we proposed in [3] does not rely on estimating parameters, and can be used to solve the stochastic NEFK problem in both static and dynamic settings. Indeed, empirical results verify that the LAKG finds the optimal solution with arbitrary accuracy, guided by the principle of Lagrange Multipliers. Furthermore, the empirical results show that the performance of the LAKG is superior to that of parameter-estimation-based schemes, both in static and dynamic environments. Accordingly, we believe that the LAKG can be considered to represent the state-of-the-art when it concerns the stochastic NEFK problem. This landmark is now extended to develop the TRAA, and its hierarchical version, the H-TRAA.

Contributions of This Paper: The contributions of this paper are the following: (1) We report the first *analytical* results for schemes that solve the optimal sampling problem using a formal asymptotically optimal solution to the Stochastic NEFK Problem. (2) We propose a novel scheme for the *two-set* sampling problem, namely, the *Twofold Resource Allocation Automaton (TRAA)*. As mentioned, from the perspective of LA, the TRAA, in itself, is the first reported LA which is *artificially* rendered ergodic. (3) We provide convergence results for the TRAA. (4) We report the first *hierarchical* solution to the Stochastic NEFK Problem, based on a hierarchy of TRAAs, namely, the H-TRAA. (5) We verify empirically that the H-TRAA provides orders of magnitude faster convergence when compared to the LAKG, and demonstrate its power to solve the underlying sampling problem.

2 A Hierarchy of Twofold Resource Allocation Automata

In order to put our work in the right perspective, we start this section by providing a brief review of the concepts found in [3] - which are also relevant for more "primitive" variants of the knapsack problem.

As indicated in the introduction, solving the classical linear FK problem involves finding the most valuable mix $\boldsymbol{x}^* = [x_1^*, \ldots, x_n^*]$ of n materials that fits within a knapsack of fixed capacity c. The material value per unit volume for each material i is given as a constant v_i, and each material is available in a certain amount $x_i \leq b_i$, $1 \leq i \leq n$. Accordingly, the value of the amount x_i of material i, $f_i(x_i) = v_i x_i$, is linear with respect to x_i. In other words, the derivative of $f_i(x_i)$ — i.e., the material value per unit volume — is fixed: $f_i'(x_i) = v_i$. Because *fractions* of materials can be placed in the knapsack, the following greedy algorithm from [1] finds the most valuable mix: *Take as much as possible of the material that is most valuable per unit volume. If there is still room, take as much as possible of the next most valuable material. Continue until the knapsack is full.*

Let us now generalize this and assume that the material unit volume values are *random* variables with *constant* and *known* distributions. Furthermore, for the sake of conceptual clarity, let us only consider binary variables that *either* instantiate to the values of 0 or 1. Since the unit volume values are random, let p_i denote the probability of the unit volume value $v_i = 1$ for material i, $1 \leq i \leq n$, which means that the probability of the unit volume value $v_i = 0$ becomes $1 - p_i$. With some insight, it becomes evident that under such conditions, the above greedy strategy can again be used to maximize the *expected* value of the knapsack, simply by selecting material based on the *expected* unit volume values, $E[v_i] = 0 \times (1 - p_i) + 1 \times p_i$, rather than actual unit volume values.

The above indicated solution is, of course, inadequate when the p_i's are unknown. Furthermore, the problem becomes even more challenging when the p_i's are no longer constant, but rather depend on their respective material amounts x_i, $1 \leq i \leq n$. Let $p_i(x_i)$ denote the probability that the current unit volume value of material i is $v_i = 1$, given that the amount x_i has already been placed in the knapsack. Then, the expected value per unit volume of material i, $1 \leq i \leq n$, becomes $E[v_i] = 0 \times [1 - p_i(x_i)] + 1 \times p_i(x_i) = p_i(x_i)$, and accordingly, the expected value of the amount x_i becomes $f_i(x_i) = \int_0^{x_i} p_i(u)du$.

Our aim, then, is to find a scheme that moves towards optimizing the following NEFK problem on-line:

$$\begin{array}{l}\text{maximize } f(\boldsymbol{x}) = \sum_1^n f_i(x_i), \text{where} f_i(x_i) = \int_0^{x_i} p_i(u)du, \text{ and } p_i(x_i) = f_i'(x_i),\\ \text{subject to } \sum_1^n x_i = c \text{ and } \forall i \in \{1, \ldots, n\}, x_i \geq 0.\end{array}$$

Note that we allow only instantiations of the material values per unit volume to be observed. That is, each time an amount x_i of material i is placed in the knapsack, an instantiation v_i at x_i is observed.

Because of the above intricacies, we approach the problem by relying on informed material mix *guesses*, i.e., by experimenting with different material mixes and learning from the resulting random unit volume value outcomes. We shall assume that x_i is any number in the interval $(0, 1)$. The question of generalizing this will be considered later. The crucial issue that we have to address, then, is that of determining how to change our current guesses on $x_i,\ 1 \leq i \leq n$. We shall attempt to do this in a discretized manner by subdividing the unit interval into N points $\{\frac{1}{N+1}, \frac{2}{N+1}, \ldots, \frac{N}{N+1}\}$, where N is the resolution of the learning scheme. It turns out that a larger value of N ultimately implies a more accurate solution to the knapsack problem.

2.1 Details of the TRAA Solution

We first present our LA based solution to *two-material* Stochastic NEFK Problems. The two-material solution forms a critical part of the hierarchic scheme for multiple materials that is presented subsequently. Our solution to two-material problems constitutes of the following three modules.

Stochastic Environment: The *Stochastic Environment* for the two-material case can be characterized by: (1) The capacity c of the knapsack, and (2) two material unit volume value probability functions $p_1(x_1)$ and $p_2(x_2)$. In brief, if the amount x_i of material i is suggested to the Stochastic Environment, the Environment replies with a unit volume value $v_i = 1$ with probability $p_i(x_i)$ and a unit volume value $v_i = 0$ with probability $1 - p_i(x_i)$, $i \in \{1, 2\}$. It should be emphasized that to render the problem both interesting and non-trivial, we assume that $p_i(x_i)$ is unknown to the TRAA.

Twofold Resource Allocation Automaton: The scheme which attempts to learn the optimal allocation $\boldsymbol{x}^* = [x_1^*, x_2^*]$ can be described as follows. A finite fixed structure automaton with the states $s(t) \in \{1, 2, \ldots, N\}$ is used to decide the allocation of resources among the two materials. Let the current state of the automaton be $s(t)$. Furthermore, let $q_{s(t)}$ refer to the fraction $\frac{s(t)}{N+1}$, and let $r_{s(t)}$ refer to the fraction: $1 - q_{s(t)}$. Then the automaton's current guess is $\boldsymbol{x} = [q_{s(t)}, r_{s(t)}]$. If the Stochastic Environment tells the automaton that the unit volume value of material i is $v_i(t)$ at time t, the automaton updates its state as follows:

$$\begin{array}{lll} s(t+1) := s(t) + 1 & \textbf{If} & \textbf{rand}() \leq r_{s(t)} \textbf{ and } v_i(t) = 1 \quad (1) \\ & & \textbf{and } 1 \leq s(t) < N \textbf{ and } i = 1 \\ s(t+1) := s(t) - 1 & \textbf{If} & \textbf{rand}() \leq q_{s(t)} \textbf{ and } v_i(t) = 1 \quad (2) \\ & & \textbf{and } 1 < s(t) \leq N \textbf{ and } i = 2 \\ s(t+1) := s(t) & \textbf{Otherwise} & \qquad\qquad\qquad\qquad\quad (3). \end{array}$$

Notice how the stochastic state transitions are designed to offset the learning bias introduced by accessing the materials with frequencies proportional to $\boldsymbol{x} = [q_{s(t)}, r_{s(t)}]$. Also observe that the overall learning scheme does not produce any absorbing states, and is, accordingly, ergodic supporting dynamic environments.

Finally, after the automaton has had the opportunity to change its state, it provides the output $\boldsymbol{x} = [q_{s(t+1)}, r_{s(t+1)}]$ to an EDF Scheduler. That is, it outputs the material amounts that have been changed.

Earliest Deadline First (EDF) Scheduler: The EDF Scheduler takes material amounts $\boldsymbol{x} = [x_1, \ldots, x_n]$ as its input (for the two-material case the input is $\boldsymbol{x} = [x_1, x_2]$). The purpose of the Scheduler is to: (1) provide accesses to the Stochastic Environment in a sequential manner, and (2) make sure that the unit volume value functions are accessed with frequencies proportional to $\boldsymbol{x}$.

The reader should note that our scheme does not rely on accessing the unit volume value functions sequentially with frequencies proportional to $\boldsymbol{x}$ for solving the knapsack problem. However, this restriction is obviously essential for solving the problem *incrementally* and *on-line* (or rather in a "real-time" manner).

Lemma 1. *The material mix $\boldsymbol{x} = [x_1, \ldots, x_n]$ is a solution to a given Stochastic NEFK Problem if (1) the derivatives of the expected material amount values are all equal at $\boldsymbol{x}$, (2) the mix fills the knapsack, and (3) every material amount is positive, i.e.:*

$$f_1'(x_1) = \cdots = f_n'(x_n)$$
$$\textstyle\sum_1^n x_i = c \textit{ and } \forall i \in \{1, \ldots, n\}, x_i \geq 0.$$

The above lemma is based on the well-known principle of Lagrange Multipliers [7], and its proof is therefore omitted here for the sake of brevity. We will instead provide our main result for the *two-material* problem and the TRAA. For the two-material problem, let $\boldsymbol{x}^* = [x_1^*, x_2^*]$ denote a solution, as defined above. Note that since x_2^* can be obtained from x_1^*, we will concentrate on finding x_1^*.

Theorem 1. *The TRAA solution scheme specified by (1)–(3) is asymptotically optimal. I.e., as the resolution, N, is increased indefinitely, the expected value of the TRAA output, $x_1(t)$, converges towards the solution of the problem, x_1^*:*

$$\lim_{N \to \infty} \lim_{t \to \infty} E[x_1(t)] \to x_1^*.$$

Proof. The proof is quite involved and is found in [8]. It is omitted here in the interest of brevity. □

2.2 Details of the H-TRAA Solution

In this section we propose a hierarchical scheme for solving n-material problems. The scheme takes advantage of the TRAA's ability to solve two-material problems asymptotically, by organizing them hierarchically.

2.2.1 H-TRAA Construction

The hierarchy of TRAAs, which we hereafter will refer to as H-TRAA, is constructed as follows[1]. First of all, the hierarchy is organized as a balanced binary tree with depth $D = \log_2(n)$. Each node in the hierarchy can be related to three

[1] We assume that $n = 2^\gamma, \gamma \in \mathbb{N}^+$, for the sake of clarity.

entities: (1) a set of materials, (2) a partitioning of the material set into two subsets of equal size, and (3) a dedicated TRAA that allocates a given amount of resources among the two subsets.

- **Root Node:** The hierarchy root (at depth 1) is assigned the complete set of materials $S_{1,1} = \{1, \ldots, n\}$. These n materials are partitioned into two disjoint and exhaustive subsets of equal size: $S_{2,1}$ and $S_{2,2}$. An associated TRAA, $T_{1,1}$, decides how to divide the full knapsack capacity c (which, for the sake of notational correctness will be referred to as $c_{1,1}$) among the two subsets. That is, subset $S_{2,1}$ receives the capacity $c_{2,1}$ and subset $S_{2,2}$ receives the capacity $c_{2,2}$, with $c_{2,1} + c_{2,2} = c_{1,1}$. Accordingly, *this* TRAA is given the power to prioritize one subset of the materials at the expense of the other.
- **Nodes at Depth d:** Node $j \in \{1, \ldots, 2^{d-1}\}$ at depth d (where $1 < d \leq D$) refers to: (1) the material subset $S_{d,j}$, (2) a partitioning of $S_{d,j}$ into the subsets $S_{d+1,2j-1}$ and $S_{d+1,2j}$, and (3) a dedicated TRAA, $T_{d,j}$. Observe that since level $D+1$ of the H-TRAA is non-existent, we use the convention that $S_{D+1,2j-1}$ and $S_{D+1,2j}$ refer to the primitive materials being processed by the leaf TRAA, $T_{D,j}$. Assume that the materials in $S_{d,j}$ has, as a set, been assigned the capacity $c_{d,j}$. The dedicated TRAA, then, decides how to allocate the assigned capacity $c_{d,j}$ among the subsets $S_{d+1,2j-1}$ and $S_{d+1,2j}$. That is, subset $S_{d+1,2j-1}$ receives the capacity $c_{d+1,2j-1}$ and subset $S_{d+1,2j}$ receives the capacity $c_{d+1,2j}$, with $c_{d+,2j-1} + c_{d+1,2j} = c_{d,j}$.

At depth D, then, each individual material can be separately assigned a fraction of the overall capacity by way of recursion, using the above allocation scheme.

2.2.2 Interaction of H-TRAA with EDF Scheduler and Environment

As in the single TRAA case, H-TRAA interacts with an EDF Scheduler, which suggests which unit volume value function $p_i(x_i)$ to access next. A response is then generated from the Stochastic Environment using $p_i(x_i)$. This response is given to all the TRAAs that were involved in determining the material amount x_i, that is, the TRAAs in the hierarchy that have allocated capacacity to a material subset that contains material i. Finally, a new candidate material mix $\boldsymbol{x} = [x_1, \ldots, x_n]$ is suggested by the H-TRAA to the EDF Scheduler.

2.2.3 Analysis of the H-TRAA Solution

In the previous section we stated the asymptotic optimality of the individual TRAA. We will now consider the H-TRAA and its optimality. In brief, it turns out that when each individual TRAA in the hierarchy has solved its own two-material problem, a solution to the complete n-material Knapsack Problem has also been produced.

Theorem 2. *Let $T_{d,j}$ be an arbitrary TRAA at level d of the H-TRAA associated with the node whose index is j. Then, if every single TRAA, $T_{d,j}$, in the H-TRAA has found a local solution with proportions $c_{d+1,2j-1}$ and $c_{d+1,2j}$ satisfying $f'_{d+1,2j-1}(c_{d+1,2j-1}) = f'_{d+1,2j}(c_{d+1,2j})$, the overall Knapsack Problem involving n materials that are hierarchically placed in $\log_2 n$ levels of TRAAs, also attains the global optimum solution.*

Proof. The proof of this theorem is also quite deep. It is included in [8], and omitted here due to space limitations. □

3 Experiments: Optimal Sample Size Determination

In this section we consider the problem of estimating the proportion of a population having some specific characteristic. Specifically, we assume that n populations are to be evaluated, and that each population i is characterized by an independent unknown binomial proportion u_i. We will here pursue the goal of minimizing the variance of the proportion estimates when the total number of samples available for estimating the proportions is restricted to c. The purpose is to make the estimates as accurate as possible. As mentioned earlier, for instance, the task at hand could be to determine the proportion of a web site that is successfully validated by an HTML validator [9], and that n web sites are to be evaluated by only accessing c web pages.

3.1 Problem Specification

Let x_i be the number of elements sampled randomly from population i and let the count Y_i be the number of the sampled elements that possess a chosen characteristic. For large x_i and when u_i is not too near 0 or 1, the estimator $\hat{u_i} = \frac{Y_i}{x_i}$ is approximately normal with mean u_i and standard deviation $s_i = \sqrt{\frac{u_i(1-u_i)}{x_i}}$ [10]. As seen, the standard deviation can be reduced (and the estimate accuracy increased) by increasing the number of samples x_i. In the problem targeted in this section, n different populations can be sampled c times and the goal is to distribute the samples among the populations to minimize the aggregated variance of the estimates. The problem can be reformulated as follows:

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} -\frac{u_i(1-u_i)}{x_i} \\ \text{subject to} & \sum x_i = c \\ 1 \le x_i,\ i = 1, \ldots, n. & \end{array}$$

The above optimization problem is an NEFK problem with concave and separable objective function. Since the u_i's are assumed unknown, we apply our H-TRAA to find a near-optimal solution incrementally and online.

3.2 The LAKG Solution

We must first define the Stochastic Environment that the LAKG is to interact with. That is, we must define the stochastic functions $\mathcal{F}' = \{f_1'(x_1), f_2'(x_2), \ldots, f_n'(x_n)\}$. By applying the principles of Lagrange multipliers we find the following conditions that characterize the optimal solution:

$$\begin{array}{l} \frac{u_1(1-u_1)}{{x_1}^2} = \cdots = \frac{u_n(1-u_n)}{{x_n}^2} \\ \sum x_i = c \\ 1 \le x_i,\ i = 1, \ldots, n \end{array}$$

Accordingly, we define $f_i'(x_i)$ as follows. First of all, each time $f_i'(x_i)$ is accessed by the LAKG, population i is sampled once and the proportion estimate $\hat{u}_i$ is updated accordingly[2]. After $\hat{u}_i$ has been updated, we instantiate $f_i'(x_i)$ by a random draw — $f_i'(x_i)$ is instantiated to the value 0 with probability $1 - \frac{\hat{u}_i(1-\hat{u}_i)}{{x_i}^2}$ and to the value 1 with probability $\frac{\hat{u}_i(1-\hat{u}_i)}{{x_i}^2}$. In other words, we keep running estimates of the u_i's in order to calculate the outcome probabilities of the $f_i'(x_i)$'s[3].

3.3 The H-TRAA Solution

In this case too, we must first define the Stochastic Environment that the H-TRAA is to interact with. Without much ado, we state that the stochastic functions $\mathcal{F}' = \{f_1'(x_1), f_2'(x_2), \ldots, f_n'(x_n)\}$, and the solutions obtained by the Lagrange multipliers leads to the *same* optimal solution as in the LAKG described in Section 3.2. The sampling philosophy, running estimates and the windowing methodology used for dynamic environments are also identical to the ones described in Section 3.2.

The H-TRAA can be configured by various means. First of all, the material amount space $(0, 1)$ need not be discretized uniformly. Instead, a nonlinear material amount space can be formed, as done for the LAKG in [3]. Furthermore, the discretization resolution N must also be set for each TRAA, possibly varying from TRAA to TRAA in the hierarchy. In short, the performance achieved for a particular problem can be optimized using these different means of configuring the H-TRAA. In this section, however, our goal is to evaluate the overall performance of the H-TRAA, without fine tuning. Therefore, we will only use a linear material amount space, as specified in Section 2. Furthermore, we will use the same resolution $N = 500$ for all the TRAAs in the hierarchy, independent of the specific knapsack problem at hand. Thus, our aim is to ensure a fair comparison with the present state of the art, namely, the LAKG scheme.

3.4 Empirical Results

3.4.1 Experimental Set-Up

In this sub-section we evaluate our learning scheme by comparing it with the optimal and uniform policies using synthetic data. The reader should appreciate that, in practice, we can only apply the uniform policy, because the optimal policy requires that the u_i's are known.

The data used in the experiment is summarized in Table 1. The table shows the true population proportions used, and the number of populations associated with each proportion. The experiment encompasses 512 populations, and the corresponding proportions are to be estimated by allocating 50,000 samples.

[2] For a dynamic environment we would utilize a "window-based" strategy and only use the last c samples to estimate the u_i's. However, we are currently studying how recently proposed weak estimators can be used in this setting [11].

[3] Because the outcome probabilities are always available for the populations, we can normalize the outcome probabilities to speed up convergence.

Table 1. The true population proportions used in the experiment, and the number of populations associated with each proportion

True Proportion	Populations
0.5	6
0.750 / 0.250	5
0.900 / 0.100	41
0.990 / 0.010	51
0.999 / 0.001	409

3.4.2 Static Environments

We first report the results obtained for environments which are static i.e., where the u_i's, although unknown, do not change with time. As we will see in the following, it turns that one of the strengths of the H-TRAA is its ability to take advantage of so-called spatial dependencies among materials. As mentioned earlier, in the above experimental setup, materials are spatially related in the sense that the updating probabilities decreases with the rank-index k. In order to starve the H-TRAA from this information, we opted to perturb this spatial structure. Each perturbation swapped the probabilities of a randomly selected material and the material succeeding it in the ranking.

We conducted our experiments with 10^3, 10^4, 10^5 and 10^6 perturbations. For each of these values, an ensemble of several independent replications with different random number streams was performed so as to minimize the variance of the reported results. The results of our experiments are truly conclusive and confirm the power of the H-TRAA. Although several experiments were conducted using various setting for various numbers of automata, we report, in the interest of brevity, a brief overview of the results obtained.

Fig. 1 plots the variance of the current solution (as a function of time) each time a unit volume value function $f_i'(x_i)$ has been sampled. The graphs show the results of applying the H-TRAA with $5,000$ states and the LAKG with $12,500$ states (where the amount of the material added on a transition in the latter is not fixed but varying in a nonlinear manner[4]).

As seen in the figure, the H-TRAA steadily reduces the variance of the initial solution in which the populations are sampled uniformly. Indeed, even by the first $50,000$ samples, one can observe a very *significant* reduction. The reader should notice that the H-TRAA converges to a near optimal allocation more expediently and far quicker than the LAKG-scheme, expect for the case with $1,000,000$ perturbations where the H-TRAA initally converges faster but subsequently in a more conservative manner.

3.4.3 Scalability

One of the motivations for designing the H-TRAA was the improved scalability by means of hierarchical learning. As seen in Fig. 2, extending the number of materials increases the convergence time of the LAKG. The H-TRAA, however, is far less affected by the number of materials.

[4] The details of this are omitted. They can be found in [3].

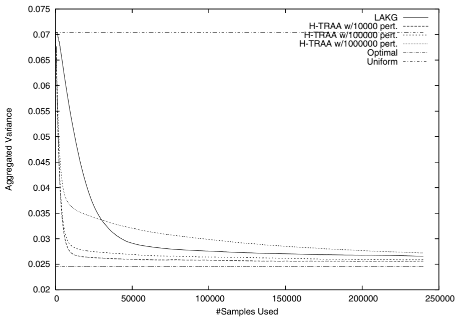


Fig. 1. The performance of LAKG and H-TRAA in the static environment

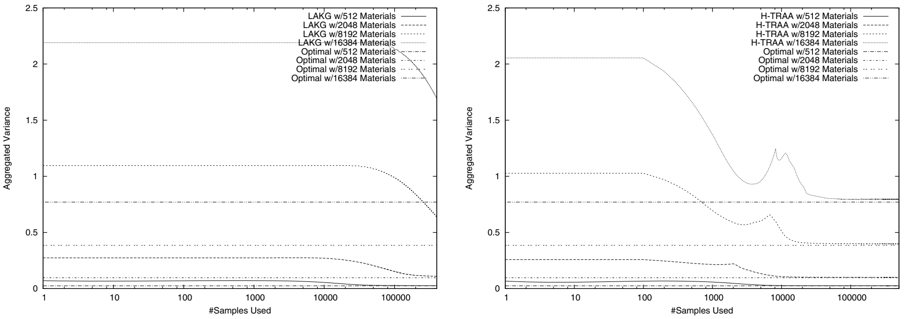


Fig. 2. Scalability of LAKG and H-TRAA

From Fig. 2 we observe that while the LAKG does not even converge, (see the figure on the left), the H-TRAA scales sub-linearly *in every case* with the number of materials (the figure on the right). However, the most interesting phenomenon that we observe from Fig. 2 is the ability of the H-TRAA to emerge out of local optima. The H-TRAA first decreases to a minimum, but when it "discovers" that there is a better solution (which in this cases implies a superior partitioning of the nodes in the tree to their left and right subtrees), it is capable of unlearning the inferior configuration and converging to a superior solution. This, we believe, is quite remarkable, especially because the size of the underlying tree is very large, implying that the number of possible binary trees (which grows exponentially with the size), is even larger. However, by arriving at the global optimum, we see that the H-TRAA has succeeded in learning the best tree structure to resolve the sampling proportions!

4 Conclusions

In this paper, we considered the problem of allocating limited sampling resources in a "real-time" manner with the purpose of estimating multiple binomial proportions. This problem is particularly intriguing because the sampling resources can only be allocated optimally if the binomial proportions are already known.

Using the general LA philosophy as a paradigm to tackle this real-life problem, our scheme improves a current solution in an online manner, through a series of informed guesses. To solve the problem, we first modelled it as a *Stochastic Non-linear Fractional Knapsack Problem.* We then presented a completely new on-line Learning Automata (LA) system, namely, the *Hierarchy of Twofold Resource Allocation Automata* (H-TRAA), whose primitive component is a *Twofold Resource Allocation Automaton* (TRAA). Both the TRAA and the H-TRAA have been proven to be asymptotically optimal.

Comprehensive experimental results demonstrated that performance of the H-TRAA is superior to the previous state-of-the-art scheme, namely, the LAKG. Finally, we also provided empirical evidence to show that the H-TRAAs possess a sub-linear scaling property, capable of emerging out of local optima.

References

1. Black, P.E.: Fractional knapsack problem. Dictionary of Algorithms and Data Structures (2004)
2. Kellerer, H., Pferschy, U., Pisinger, D.: Knapsack Problems. Springer (2004)
3. Granmo, O.C., Oommen, B.J., Myrer, S.A., Olsen, M.G.: Learning Automata-based Solutions to the Nonlinear Fractional Knapsack Problem with Applications to Optimal Resource Allocation. IEEE Transactions on Systems, Man, and Cybernetics, Part B **37**(1) (2007) 166–175
4. Pandey, S., Ramamritham, K., Chakrabarti, S.: Monitoring the Dynamic Web to Respond to Continuous Queries. In: 12th International World Wide Web Conference, ACM Press (2003) 659–668
5. Narendra, K.S., Thathachar, M.A.L.: Learning Automata: An Introduction. Prentice Hall (1989)
6. Tsetlin, M.L.: Automaton Theory and Modeling of Biological Systems. Academic Press (1973)
7. Bretthauer, K.M., Shetty, B.: The Nonlinear Knapsack Problem — Algorithms and Applications. European Journal of Operational Research **138** (2002) 459–472
8. Granmo, O.C., Oommen, B.J.: Learning automata-based solutions to optimal sampling by solving a nonlinear fractional knapsack problem (2008) Unabridged version of this paper. Submitted for publication.
9. Snaprud, M., Ulltveit-Moe, N., Granmo, O.C., Rafoshei-Klev, M., Wiklund, A., Sawicka, A.: Quantitative Assessment of Public Web Sites Accessibility - Some Early Results. In: The Accessibility for All Conference. (2003)
10. Bhattacharyya, G.K., Johnson, R.A.: Statistical Concepts and Methods. John Wiley & Sons (1977)
11. Oommen, B.J., Rueda, L.: Stochastic Learning-based Weak Estimation of Multinomial Random Variables and its Applications to Pattern Recognition in Non-stationary Environments. Pattern Recognition (2006)

A Human Gait Classification Method Based on Adaboost Techniques Using Velocity Moments and Silhouette Shapes

Chin-Shyurng Fahn, Ming-Jui Kuo, and Min-Feng Hsieh

Department of Computer Science and Information Engineering,
National Taiwan University of Science and Technology,
Taipei, Taiwan 10607, Republic of China
{csfahn,D9515007,M9315023}@mail.ntust.edu.tw

Abstract. In this paper, we propose a human gait classification method based on Adaboost techniques that identify three kinds of human gaits: walk, run, and limp. We divide a video sequence into several segments, each of which is regarded as a process unit. For each process unit, we collect both the velocity and shape information of a moving object. We first apply the Canny edge detector to enhancing the edges within a foreground image, followed by computing the distance and the angle difference between two edge pixels which are put into the accumulation table. The gait classification employs an Adaboost algorithm which is excellent in facilitating the speed of convergence during the training. The experimental result reveals that our method has good performance of classifying human gaits using both the velocity and shape information.

Keywords: Gait classification, velocity moment, silhouette shape, Canny edge detector, accumulation table, Adaboost algorithm.

1 Introduction

Visual surveillance is a hot topic in recent years, since it can accomplish many tasks for human being. Visual surveillance systems can be applied in the field of security service, intelligent transportation, homecare, and so forth. They play an important role in our life, especially when the security service is paid more attention by people. The functionality of these systems is based on automated human behavior analysis. More researches have been devoted to study such a topic. Human gait classification has two major categories of approaches. That is, template matching or stochastic processing. Each of them extracts the gait information from video sequences to constitute feature vectors. For template matching, Shutler et al. developed new velocity moments in automatic gait recognition [1]. Moreover, Tassone et al. used a point distribution model to recognize walking and running at different speeds for a person on a treadmill [2].

No matter what classification techniques are adopted, the correction rate is mainly determined by the choice of features. The more robust the features stand for, the higher likelihood of distinguishing different human gaits obtains. The velocity

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 535–544, 2009.

information provides the primary differences of varied human gaits. In literature [1,2], the used features are strongly related to the velocity of a moving object. Although their performances are well, the recognition results are easily influenced by the factor of velocities. The shape information is not like the velocity information which is obviously distinguishable in various kinds of human gaits. So they apply mathematical models to analyze the features of human gaits and take more processing time. In this paper, we use both velocity and shape information as the features to develop a human gait classification system which can distinguish three kinds of gaits: walk, run, and limp. In the main, the velocity of running is faster than that of walking. Additionally, the poses of upper and lower limbs would have tiny differences. Therefore, we exploit both velocity moments and silhouette shapes as pose features for classifying human gaits by means of Adaboost techniques. Figure 1 shows the flow chart of our human gait classification system.

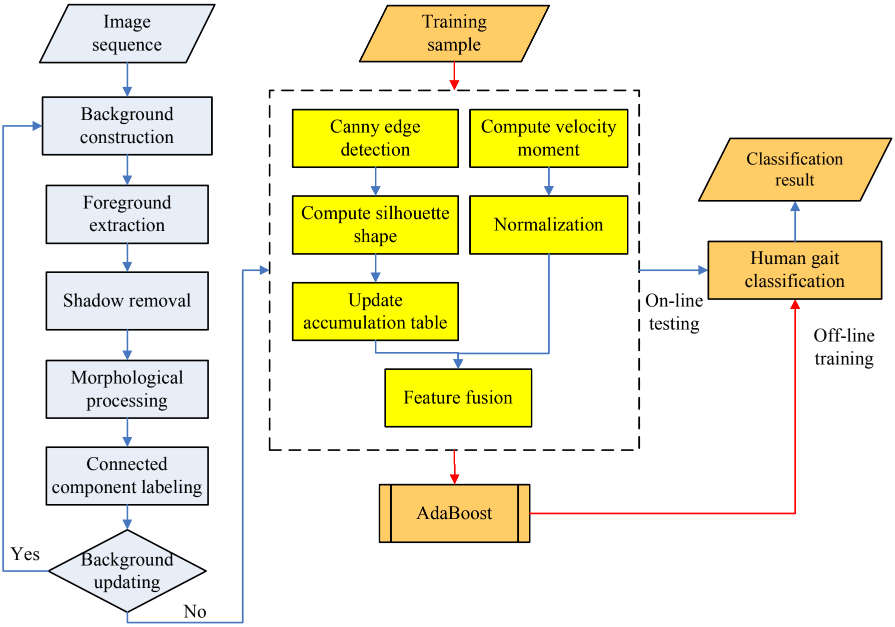


Fig. 1. The system flow chart of human gait classification

2 Preliminary Processing

Before classifying human gaits, we would perform some preliminary processes on input video sequences. That is helpful to the subsequent processing such as feature extraction. At first, we introduce background construction and how to extract a moving object. Then we describe a background updating method and a shadow removal technique to reduce the shadow effect. We also apply morphological operations on the resulting image in advance of connected component labeling to obtain desired foregrounds.

2.1 Background Construction and Foreground Extraction

In a video sequence, we are only interested in the moving object constituting a foreground. In general, the change of brightness is derived by subtracting the pixel value of the background image from that of the current image. We can set a threshold to tell whether the pixel belongs to a foreground or background. However, the setting of a threshold needs a lot of considerations, especially in a low contrast case. Kuno et al. proposed an extraction function that arranges the sensitivity of the differences between the background and the current images according to the brightness of each pixel [3]. We employ this extraction function to find foregrounds. The extraction function is expressed as follows:

$$f(a,b) = 1 - \frac{2\sqrt{(a+1)(b+1)}}{(a+1)+(b+1)} \cdot \frac{2\sqrt{(256-a)(256-b)}}{(256-a)+(256-b)} \tag{1}$$

where a is the brightness in the current image for a certain pixel and b is the brightness in the background image for the corresponding pixel.

2.2 Morphological Image Processing and Shadow Removal

After moving objects are extracted, we further perform morphological operations on them to get better quality. The erosion and dilation operations are both the conventional and popular methods used to clean up anomalies in the objects. The combination of these two operations can also achieve good performance. For example, using a dilation operation followed by an erosion operation can remove small holes in the objects and smooth out interlacing anomalies. Furthermore, the shadow is easily misclassified as a foreground at the background subtraction stage. Therefore, the foreground derived from background subtraction may not be our desired object. Sometimes the unexpected object would affect the subsequent processing result. Unfortunately, no method is effective to remove shadows. But there are some existing methods to reduce the influence of shadows. We adopt the method that was proposed by Cucchiara et al., since it is stable in various situations [4].

The reason why we employ the *HSV* color space is the human's perception tending to such one that has good discrimination between object's shadows and the background. In fact, the hue value of a shadow pixel is slightly different from that of a background pixel. This means the brightness has little effect on the hue value. Additionally, the saturation value is taken into account because the part of a shadow usually has low saturation. Shadow removal is executed using Eq. (2).

$$SP_k(x,y) = \begin{cases} 1, & \text{if } \alpha \le I_k^V(x,y) / B_k^V(x,y) \le \beta \text{ , } (I_k^S(x,y) - B_k^S(x,y)) \le \tau_S \\ & \text{and } \left| I_k^H(x,y) - B_k^H(x,y) \right| \le \tau_H \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $I_{k(x,y)}$ and $B_{k(x,y)}$ are the pixel values of the k-th image and the background image, respectively. The superscripts H, S, and V stand for the three components of the *HSV* color space. Both α and β are the ranges of the ratio of the intensity components; τ_s and τ_H are the thresholds of the differences of the saturation and hue components, individually. After the shadow removal process, we apply a connected component labeling method to make the foreground pixels connect.

3 Feature Extraction and Gait Classification

Once the preliminary processing has been accomplished, we will extract the features from the processed foreground image. For gait classification, we choose the velocity moments and silhouette shape of a moving object as its pose features. A strong classifier for recognizing human gaits is then realized by an Adaboost algorithm through a cross-validation scheme.

3.1 Velocity Information

Moments are broadly applied in pattern recognition and image processing, which describe the content of an image with respect to its axes. Two main moment approaches are the Zernike moment [5] and geometric moment [6]. In this paper, we exploit velocity moments to characterize the velocity information of a moving object, which originates from the geometric moment.

1. Geometric moment: The geometric moment is also called Cartesian moment. The $(p+q)$-th order Cartesian moment of a 2-D continuous function $p(x,y)$ is defined below:

$$m^*_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q p(x,y)dxdy \tag{3}$$

for a digital image p_{xy}, the $(p+q)$-th order Cartesian moment could be defined as

$$m_{pq} = \sum_{x=1}^{M}\sum_{y=1}^{N} x^p y^q P_{xy} \tag{4}$$

where M and N are the dimensions of the image and $x^p y^p$ is the basis function. The zero-th order moment m_{00} is the total mass of an image, which is stated in Eq. (5).

$$m_{00} = \sum_{x=1}^{M}\sum_{y=1}^{N} P_{xy} \tag{5}$$

From the above equation, we could find that the zero-th order moment m_{00} is just a pixel count; that is, the number of pixels of an image. The first order moments are usually to represent the center of mass of the image. It is defined as the centroid whose coordinates can be given as Eq. (6).

$$\bar{x} = m_{10}/m_{00} \text{ and } \bar{y} = m_{01}/m_{00} \tag{6}$$

2. Velocity moment: As its name stipulates, the velocity moment takes account of velocity factors. With reference to [1], the Cartesian velocity moments (VMs) are computed from a sequence of I consecutive images as Eq. (7).

$$vm_{pq\mu r} = \sum_{i=2}^{I}\sum_{x=1}^{M}\sum_{y=1}^{N} C_i \cdot V_i \cdot P_{i_{xy}} \tag{7}$$

where C_i symbolizes the centralized moments as

$$C_i = (x_i - \overline{x}_i)^p (y_i - \overline{y}_i)^q \tag{8}$$

and V_i indicates the velocity as

$$V_i = (\overline{x}_i - \overline{x}_{i-1})^\mu (\overline{y}_i - \overline{y}_{i-1})^\gamma \tag{9}$$

In the above two equations, $\overline{x}_i$ and $\overline{x}_{i-1}$ are the centroids in the x-direction of the (i-1)-th and i-th images, respectively. Both $\overline{y}_i$ and $\overline{y}_{i-1}$ are the centroids in the y-direction of the (i-1)-th and i-th images, respectively.

The (p+q)-th order VMs for $\mu = 0$ and $\gamma = 0$ mean the summation of the centralized moments as shown below:

$$vm_{pq00} = \sum_{i=2}^{I} \sum_{x=1}^{M} \sum_{y=1}^{N} (x_i - \overline{x}_i)^p (y_i - \overline{y}_i)^q P_{ixy} \tag{10}$$

In a similar way, for $p = 0$ and $q = 0$, the (μ+γ)-th VMs would become the summation of the differences between the centers of mass of successive images (i.e., velocity) as Eq. (11) shows:

$$vm_{00\mu\gamma} = \sum_{i=2}^{I} \sum_{x=1}^{M} \sum_{y=1}^{N} (\overline{x}_i - \overline{x}_{i-1})^\mu (\overline{y}_i - \overline{y}_{i-1})^\gamma P_{ixy} \tag{11}$$

In practice, the results of VMs are normalized with respect to the number of images and the average area of moving objects. The normalization is expressed as Eq. (12):

$$\overline{vm}_{pq\mu\gamma} = vm_{pq\mu\gamma} \big/ A \cdot I \tag{12}$$

where A is the average area of moving objects and I is the number of images.

Distinct kinds of human gaits may have different velocities. Generally, the speed of walking is slower than that of running. And the speed of limping is slower than that of walking. Therefore, the speed can be taken as a pose feature to classify human gaits.

3.2 Shape Information

Besides the velocity information, we extract the shape information of moving objects as the pose feature for human gait classification. As the image is binary, we perform the Canny edge detection to get the boundary of a foreground. The resulting silhouette shape implies the other information of classifying human gaits.

Vega and Sarkar proposed a feature extraction method [7], and we adopt a similar manner to extract shape features. As the border information is acquired, we start to extract the shape features from the silhouette of a foreground. At the beginning, we choose a foreground pixel as an initial point. Then the distance d between the initial point and another foreground pixel is computed, which acts as a type of shape features for human gait classification. Since the border information is derived from the Canny edge detection algorithm, each foreground pixel has its angle value which represents the orientation of that constituting edge. Figure 2 illustrates the distance and the angle difference between two edge pixels. In this calculation, the distance d would be divided by the height H of that constituting silhouette for normalization, so the distance becomes d/H and it ranges from 0 to 1. We further classify the distance as five regions: [0, 0.2), [0.2, 0.4), [0.4, 0.6), [0.6, 0.8), and [0.8, 1.0]. In the similar way, the angle difference φ would be classified as four regions, such as 0°, 45°, 90°, and 135°. In practice, we would create a table consisting of quantization cells to accumulate the number of foreground pixels.

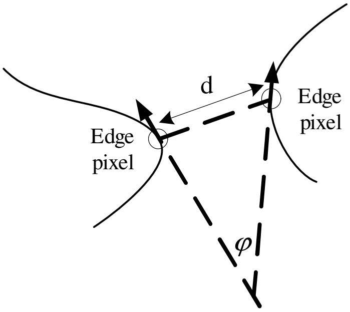


Fig. 2. The geometric relationship between two edge pixels

3.3 Classification Strategies

Boosting is a general method for enhancing the performance and extending the capabilities of a learning scheme. It is an additive logistic regression model of producing an accurate prediction rule by combining rough and moderately inaccurate "rules of thumb." For recognizing human gaits, we adopt an Adaboost (adaptive boosting) algorithm to realize a strong classifier which was proposed by Freund and Schapire [8]. The weak classifier is the core of an Adaboost algorithm. Each weak classifier produces the answer "yes" or "no" for a particular feature. We employ a classification and regression tree (CART) to realize the weak classifier that is further integrated with the Adaboost algorithm. The classical CART algorithm was proposed by Breiman et al. [9].

4 Experimental Results and Discussion

We will demonstrate the effectiveness of our human gait classification system by comparing the results with a single type and mixed type of features in terms of velocity moments and silhouette shapes. Our developing environment includes Borland C++ Builder 6, MATLAB 7.2 and Microsoft Windows XP; the personal computer is equipped with a Pentium IV 3.2 GHz CPU and 1 GB RAM. All the image sequences we use in experiments are taken by a digital camera whose type is Sony P8.

4.1 Experimental Scheme

Before classification, we should set up our gait database. At the training stage, rather than processing a whole sequence directly, we divide an image sequence into a series of segments, namely process units. Then we perform cross-validation on these process units and construct the classification model.

1. Process unit: In the experiment, we separate each image sequence into many process units, one of which consists of 15 frames, and we extract features from a process unit. To express the shape information, we calculate the value of an accumulation table for each frame of a process unit. In place of extracting features from each frame, we alternately partition a process unit into three parts to reduce the dimension of features. Every part has 5 frames, and we get the shape information from each part by taking an average of the values of the original accumulation tables. Therefore, each process unit has three new accumulation tables that store the shape information in different periods.

   We not only exploit shape information but also velocity information to classify human gaits. We utilize Eq. (12) to calculate three velocity moments for each process unit. In a process unit, the dimension of each accumulation table is 20; hence, the total dimension of a process unit including three accumulation tables and three velocity moments is 63. After manipulating the features necessarily, we stack them as a 63-dimensional feature vector used in the classification of human gaits.
2. Cross-validation: For training and classification, we adopt the GML Adaboost Matlab Toolbox which was provided by the Graphics and Media Laboratory of Computer Science Department at Moscow State University and Matlab 7.2 for training/constructing our strong classifier. During the training, we employ cross-validation for estimating generalization errors. In the experiment, we build different CARTs to find out the optimal numbers of training iterations, such as 8-split and 16-split CARTs in the 5-fold cross-validation manner. According to the simulation results, we choose the 8-split CART with 100 iterations to train our strong classifier, since its performance is better than the others.
3. Classification structure: We exploit AdaBoost techniques to classify human gaits, but the original AdaBoost algorithm is only used for a binary classifier. In this paper, we intend to classify three kinds of human gaits, so we must develop a classification structure to cooperate with the AdaBoost algorithm for our classification task. The structure is in form of a tree. Each leaf node stands for a kind of human gaits, and the internal node of the tree stands for the hypothesis to classify input data.

4.2 Tests on Gait Classification

In the following experiments, we employ three types of features (velocity moment, silhouette shape, and the fusion of these two) to classify human gaits and compare their performances. We make test video sequences to classify three different kinds of human gaits: walk, run, and limp. Each kind of gaits is taken in 15 sequences, and each sequence only records a single type of motions. As mentioned previously, we divide each video sequence into several process units. At the classification stage, each process unit would be given a value by the strong classifier trained with the Adaboost algorithm. This value symbolizes which kind of gaits the process unit belongs to. A video sequence is assigned to a certain kind of gaits when the major parts of those process units represent that kind. In addition to this, we set a threshold in our strong classifier to screen out the video sequence with less similar process units. If the number of the majority of process units is smaller than the threshold, the video sequence would be regarded as the "other class." Moreover, we conduct different experiments on the same test video sequences with three types of features listed below:

Type A: velocity moments (vm_{0010}, vm_{0001} , and vm_{0011});
Type B: silhouette shapes;
Type C: velocity moments and silhouette shapes.

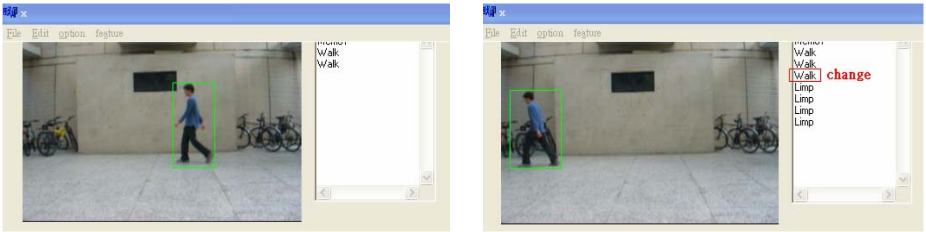
Experiment A: In this experiment, we only use one type of features: velocity moments to test. Table 1 shows the experimental result (See Part A) where the classification of "run" and "limp" motions performs well, but not the "walk" motion. It is caused by the walk motion misclassified as the run motion easily, since the velocities of run and walk motions are quite similar to each other sometimes.

Experiment B: In this experiment, we just employ one type of features: silhouette shapes to verify. Table 1 records the classification result (See Part B). Compared with the previous experiment, the performance is relatively worse. The result shows that only using shape information is not enough to classify the human gaits. It may be the process unit containing too many frames, so that the shape information is critically smoothed by averaging the constituting features. To improve the classification performance, we can increase the size of the accumulation table to store detailed shape data or extract the dominant vector from the table by means of PCA as the pose features.

Experiment C: In the experiment, we fuse two types of features: velocity moments and silhouette shapes to test. Table 1 lists the result of classifying the human gaits (See Part C). Such performance is as well as that of Experiment A, especially that the classification result of the "walk" motion is better than those in the other two experiments. It implies that considering both the velocity and shape information as the pose features can obtain a preferable accurate result. Furthermore, we can find that the classification result is influenced strongly by the velocity information. Figure 3 illustrates two exemplary frames of a multi-type motion sequence.

Table 1. Classification Results of Experiments A, B, and C

Gait type	Classification result											
	Walk			Run			Limp			Others		
	A	B	C	A	B	C	A	B	C	A	B	C
Walk	8	7	10	4	5	2	2	0	2	1	3	1
Run	3	3	0	12	9	13	0	2	0	0	1	2
Limp	3	4	1	0	6	0	12	5	14	0	0	0



Walk
Walk

Walk
Walk
Walk
Limp
Limp
Limp
Limp

(a) (b)

Fig. 3. Exemplary human gaits in a walk-to-limp motion sequence: (a) a walk motion; (b) a limp motion

5 Conclusions

In this paper, we have presented an effective method to classify human gaits. It utilizes both the velocity and shape information as the pose features fed to a strong classifier based on AdaBoost techniques. For a stationary camera, this method extracts the foreground frame by frame, so we do not require the functionality of tracking moving objects. We employ an AdaBoost algorithm to classify three kinds of human gaits: walk, run, and limp. Many experiments are made to test the video sequences with three types of features, including velocity moments, silhouette shapes, and the fusion of these two. From the experimental outcomes, using the velocity mixed with shape information achieves better results than using either of them does. The advantage of the AdaBoost algorithm is the short convergence time. It is conducive to update new types of features and add new kinds of human gaits to classify. Moreover, our proposed method can take a prescribed unit time to sequentially classify the human gaits, so it does not obtain the classification result until the whole video sequence is completely processed. Therefore, the classification system can reply what kind of human gaits in very short time.

Acknowledgement

The authors are thankful for this work supported in part by the National Science Council of Taiwan (R.O.C.) under Grant NSC95-2213-E-011-105.

References

1. Shutler, J.D., Nison, M.S., Harris, C.J.: Statistical gait description via temporal moments. In: Proceedings of the 4th IEEE Southwest Symposium on Image Analysis and Interpretation, Austin, TX, pp. 291–295 (2000)
2. Tassone, E., West, G., Venkatesh, S.: Temporal PDMs for gait classification. In: Proceedings of the 16th International Conference on Pattern Recognition, Quebec, Canada, vol. 2, pp. 1065–1068 (2002)

3. Kuno, Y., Watanabe, T., Shimosakoda, Y., Nakagawa, S.: Automated detection of human for visual surveillance system. In: Proceedings of the International Conference on Pattern Recognition, Vienna, Austria, pp. 865–869 (1996)
4. Cucchiara, R., Grana, C., Neri, G., Piccardi, M., Prati, A.: The Sakbot system for moving object detection and tracking. In: Remagnino, P., Jones, G.A., Paragios, N., Regazzoni, C.S. (eds.) Video-based Surveillance Systems: Computer Vision and Distributed Processing, pp. 145–157. Springer, Berlin (2001)
5. Zernike, F.: Beugungstheorie des schneidenverfahrens und seiner verbesserten form, der phasenkontrastmethode. Physica 1, 689–704 (1934)
6. Teague, M.R.: Image analysis via the general theory of moments. Journal of the Optical Society of America 70(8), 920–930 (1980)
7. Vega, I.R., Sarkar, S.: Statistical motion model based on the change of feature relationships: human gait-based recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(10), 1323–1328 (2003)
8. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Proceedings of the 13th International Conference on Machine Learning, Bari, Italy, pp. 148–156 (1996)
9. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Chapman and Hall, New York (1984)

A Generic Episodic Learning Model Implemented in a Cognitive Agent by Means of Temporal Pattern Mining

Usef Faghihi[1], Philippe Fournier-Viger[1], Roger Nkambou[1], and Pierre Poirier[2]

[1] Department of Computer Science, UQAM
[2] Cognitive Science Institute, UQAM
201, avenue du Président-Kennedy, Local PK 4150, Montréal (Québec)
{faghihi.usef,fournier_viger.philippe}@courrier.uqam.ca,
{nkambou.roger,poirier.pierre}@uqam.ca

Abstract. This paper describes a Conscious Tutoring System (CTS) capable of dynamic fine-tuned assistance to users. We put forth the integration of an Episodic Learning mechanism within CTS that allows it to first establish, through data mining algorithms, gross user group models. CTS then uses these models to classify incoming users, evaluate their performance, predict their future behavior, and, through a Pedagogical knowledge mechanism, decide which tutoring intervention fits best.

Keywords: Autonomous Agents, Cognitive Tutoring Agent, Episodic Memory, Emotions, Episodic Learning.

1 Introduction

Memory is very important for cognitive tutoring agents. Episodic memory is one particularly important form of memory [1]. It allows an agent online adaptation of its behavior in relation to its environment or surrounding agents; this, according to previous experiences. The agent's Episodic learning helps it to learn and then remember logged information about the behavior of learners or other agents during training sessions. It also helps the agent remember learners' previous relevant mistakes and previously used methods- whether successful or unsuccessful.

In humans, the memory of *what*, *where* and *when*, known as episodic memory, is influenced both directly and indirectly by the amygdala, which play a major role in emotional processes [2]. Recently, studies have demonstrated the role of the hippocampus and its influences on episodic memory consolidation in the human brain [5]. These suggest that learning first occurs in the hippocampus through a fast process, from where the acquired information is then transferred, via a slower process, to various cortical areas. One interesting model of the human memory consolidation phase [5] is the multiple-trace theory. It holds a hippocampus-dependent view of event encoding. Every time an event causes memory reactivation, a new trace for the activated memory is created in the hippocampus. Memory consolidation occurs through the reoccurring loops of episodic memory traces in the hippocampus and the construction of semantic memory traces in the cortex.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 545–555, 2009.

Emotions affect different types of memory and enhance learning in humans [3]. Scientists now know that we automatically create emotional valences of events in episodic memory. In fact, recent neuroimaging studies [2] show a relation between the activation of the anterior temporal lobe and emotional memory retrieval. Because emotions and episodic memory play complementary roles in human learning and retrieval phase, we believe that these should be included in cognitive architectures.

CTS is inspired by the most recent neurobiological and neuropsychological theories of human brain function (see figure 1). It is an agent designed to provide assistance to the learner during training in virtual learning environments. In our work, it is applied to a tutoring system called RomanTutor [16] in order to provide assistance to astronauts learning how to manipulate Canadarm2, the robotic telemanipulator attached to the International Space Station (ISS). Our team has now added an emotional mechanism into the CTS [4] architecture. We here explain the crucial role episodic memory (or pseudo-hippocampus) plays in our model's episodic learning mechanisms associated with emotion.

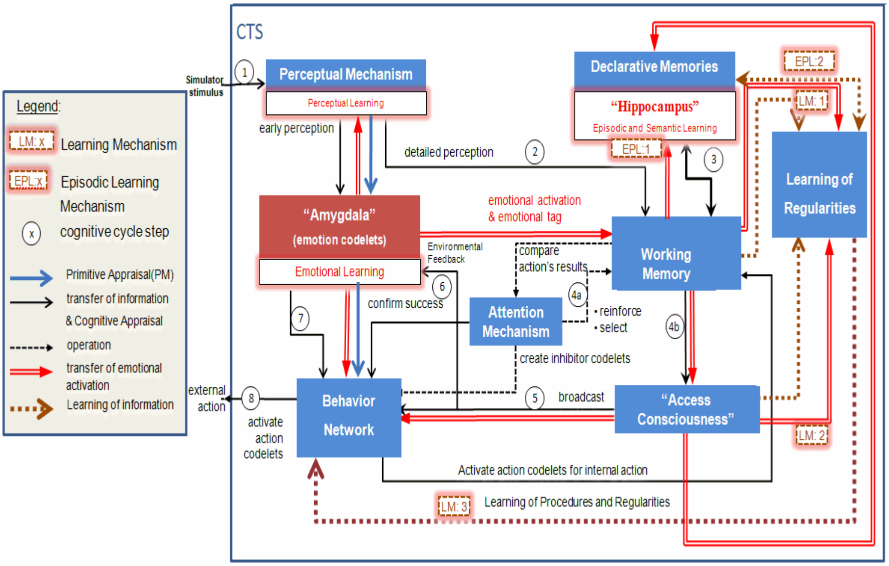


Fig. 1. CTS Architecture

Because CTS is endowed with cognitive cycles, it receives a huge amount of data while interacting with its environment, we believe our proposed model could be very important. The data, in order to be used in decision making, needs to be consolidated into smaller form. In our architecture, this is achieved through sequential pattern mining, for it is an efficient method widely used to find frequent temporal patterns among sequences of symbols. These sequential patterns are useful in the retrieval (remembering) phase to adapt CTS's behavior to past experiences. In the present paper, we will begin by a brief review of the existing work concerning episodic learning in cognitive agents. We will then propose our new architecture combining elements of the Emotional Mechanism (EM) and episodic memory. Finally, we will present results from our experiments with this cognitive agent.

2 Cognitive Agents with Episodic Memory

Many have attempted to incorporate episodic memory and learning mechanisms in cognitive agents and cognitive architectures, for example ACT-R (Sidney et al., 2006). Yet, none have included a role for emotions in the episodic learning and retrieval processes as of now. In the well-known ACT-R model, there is no explicit episodic memory. Instead, episodic events are encoded in the declarative memory as chunks, just like declarative information. During the recall, beside the activation provided by the context, a base level activation function is used for each chunk to calculate the probability of being retrieved and the speed of its retrieval. Basically, the activation is calculated based on the time elapsed since the last occurrence of the chunk in Working Memory (WM) and the number of times that the chunk was recalled. Because chunk activation decreases rapidly over time, after a short while, the frequency of chunk use becomes the most decisive feature for determining recall. Thus, ACT-R cannot recall information in a temporal context, and this induces abnormal behavior [17]. In addition, since ACT-R has no emotions, these cannot be taken into account during episodic memorization and retrieval.

3 Episodic Memory in CTS Architecture

CTS architecture relies on the functional "*consciousness*" [11] mechanism for much of its operations. It also bears some functional similarities with the physiology of the nervous system. Its modules communicate with one another by contributing information to its Working Memory through information codelets[1] (See [4] for more details). We have incorporated in CTS a form of self-sufficient learning - Episodic learning. This takes place through CTS's cognitive cycle (figure 1), of which we give a brief overview below.

Let us first quickly review Episodic learning (also called *pseudo-hippocampus*)in CTS. This mechanism consists of the encoding of any given information coupled with its assigned emotional valence, and the encoding of the agent's actions. It is composed of two main mechanisms called "memory consolidation" and "episodic learning" [5]. These intervene in the memorization and the retrieval phases of the events in CTS's memory architecture respectively. The memorization phase of the CTS architecture includes emotional valences[2] ascribed to ongoing events by the Emotional Mechanism [6]. All events sequences are distributed and then stored in

[1] Based on Hofstadter *et al.'s* idea [13], a *codelet* is a very simple agent, "a small piece of code that is specialized for some comparatively simple task". Implementing Baars theory's *simple processors*, codelets do much of the processing in the architecture. In our case, each information codelet possesses an activation value and an emotional valence specific to each cognitive cycle.

[2] Emotional valences are here taken to be CTS's memorization of valenced reactions to given emotional situations (stimuli) as described in the OCC model [7].

different memories in CTS. The memorization phase also includes the process of memory consolidation [5]. This process constantly extracts temporal regularities from all past episodes to form a procedural memory.

Two routes are possible in the cognitive cycle of CTS- a short route[3] and a long route. In both cases, the cycle begins with the perceptual mechanism. Hereafter, we briefly summarize each step in the cycle and in *italics*, describe the influence of emotions (here called *pseudo-amygdala*[4]or EM and/or of *pseudo-hippocampus* (PH). For a visual of the same, please refer to Figure 1.

Step 1: The first stage of the cognitive cycle is to perceive the environment; that is, to recognize and interpret the stimulus.

EM: All incoming information is evaluated by the emotional mechanism when low-level features recognized by the perceptual mechanism are relayed to the emotional codelets, which in turn feed activation to emotional nodes in the Behavior Network. Strong reactions from the "pseudo-amygdala" may cause an immediate reflex reaction in the CTS [6].

Step 2: The percept enters WM.

In this step, if the received information is considered important or dangerous by EM, there will be a direct reaction from EM which primes an automatic behavior from BN [6].

PH[5]: PH also *inspects and fetches WM relevant information and sends it back to WM once enriched. Relevant traces from the different memories are automatically retrieved. These will be sequences of events in the form of a list relevant to the new information. The sequences include the current event and the residual information from previous cognitive cycles in WM. The retrieved traces contain codelet links with other codelets. Each time new information codelets enter WM, the memory traces are updated depending on the new links created between these traces and the new information codelets.*

Step 3: Memories are probed and other unconscious resources contribute. All these resources react to the last few consciousness broadcasts (internal processing may take more than one single cognitive cycle).

EM: What is brought back from episodic memory is evaluated by the emotional codelets (as part of emotional intervention ELR: 2 in Figure 1) and receives anew its emotional load.

Step 4: Coalitions assemble. In the reasoning phase, coalitions of information are formed or enriched. Attention codelets join specific coalitions and help them compete with other coalitions toward entering "*consciousness*".

[3] The short route is a *percept-reaction*, direct process which takes place when the information received by the perceptual mechanism is deemed most strong by the *pseudo-amygdala*. The short route is described elsewhere [6].

[4] Let us note that in CTS, a "*pseudo-amygdala*" is responsible for emotional reactions [6].

[5] A "pseudo-hippocampus" mediates all declarative memories involved (in the long route).

EM: Emotional codelets observe the WM's content, trying to detect and instill energy to codelets believed to require it and attach a corresponding emotional tag. As a result, emotions influence which information comes to consciousness, and modulate what will be explicitly memorized.

Step 5: The selected coalition is broadcast. The Attention mechanism spots the most energetic coalition in WM and submits it to the "access consciousness," which broadcasts it to the whole system. With this broadcast, any subsystem (appropriate module or team of codelets) that recognizes the information may react to it.

PH: retrieve the past frequently reappearing information best matching the current information resident in WM. It extracts frequent partial or complete event sequences (episodic patterns) from the list of special events previously consolidated. This may invoke a stream of behaviors related to the current event, with activation passing through the links between them. These could be considered partial or complete (procedures) of actions.

Steps 6 and 7: Here unconscious behavioral resources (action selection) are recruited. Among the modules that react to broadcasts is the Behavior Network (BN). BN plans actions and, by an emergent selection process, decides upon the most appropriate act to adopt. The selected Behavior then sends away the behavior codelets linked to it.

EM: In this step when the BN starts a deliberation, for instance to build a plan, the plan is emotionally evaluated as it is built, the emotions playing a role in the selection of the steps. If the looping concerns the evaluation of a hypothesis, it gives it an emotional evaluation, perhaps from learned lessons from past experiences.

PH: Up to now, only the Behavior Network (BN) (inspired from[14]), having received these broadcasted coalitions, could plan and execute actions as well as monitor frequent partial or complete event sequences. As we have here seen, in our revised CTS model, the PH can now also do this, and does it better.

Step 8: Action execution. Motor codelets stimulate the appropriate muscles or internal processes.

EM: *Emotions influence the execution, for instance in the speed and the amplitude of the movements.*

In the next section we explain in detail how episodic learning is implemented in CTS's architecture

4 CTS's Episodic Learning

In this section we explain how Episodic Learning is implemented in the CTS's architecture. We then explain how our Episodic Memory consolidation model and Episodic learning process occur in CTS's architecture.

4.1 Episodic Learning in CTS

Episodic Learning (EPL) in CTS starts when; the information codelets entered in the WM make it so that an event during consciousness broadcast is considered an episode. These might be learned by CTS's *pseudo-hippocampus* (PH) in each cognitive cycle. As mentioned above, CTS's pseudo-hippocampus learns the traces. This learning happens through the creation of new sequences of events. Each sequence may contain one or more nodes that have links with other nodes situated in the sequence. Learning happens through the strengthening/weakening of the energy of the nodes and links between them. As regards the information entered in WM, if the PH does not have a response set for the information broadcast by the consciousness broadcast mechanism, it creates a new sequence with a unique ID. It then creates an empty node with a context which explains ongoing situations (current event). Observing all broadcast information by the consciousness mechanism, PH gives a unique ID to each coalition broadcast in the system and saves them instantaneously. To fill out each node, the EM waits for the reasoning phase, the consciously-selected behavior and the ensuing broadcasting of the externally-confirmed event. At this point, each node in the sequence is assigned the time of broadcasted coalition, its total emotional valence, and a key-information-codelet (trigger-codelet) associating to the broadcast coalition that fires the stream of BN (if it has passed its threshold value). The PH then associates the context of the new node with the ID of the broadcast coalition consciously-selected by the Attention mechanism and executed by the BN. The emotional valences corresponding to this broadcast coalition are also saved. At this point the information is ready to be integrated in the different memories of the system. The sequence(s) related to this episode are saved in a database which is considered as CTS's Episodic Memory. This as well as the information learned by CTS's Learning Mechanisms (i.e. learning of regularities [18], procedural learning and emotional learning [6] during arm manipulation) is distributed and then integrated in the same database separately from Episodic Memory. With this method, CTS clearly relates an episode to its corresponding procedures in the BN. In the next two sections we explain in detail how the episodic memory consolidation and episodic learning processes are implemented in CTS's architecture.

4.2 The Episodic Memory Consolidation Model

The memory consolidation process takes place after each of CTS's cognitive cycles. Like the human hippocampus, CTS's EPL extracts frequently occurring event sequences during arm manipulation by the astronauts in the Roman Tutor's virtual world. In our agent, an episodic trace or sequence of events is recorded during *consciousness broadcast* as mentioned in section 3. To mine frequent events sequences, we chose the sequential pattern mining algorithm of [15] which provides several more features than the original GSP sequential pattern algorithm [8], such as accepting symbols with numeric values, eliminating redundancy and handling time constraints. The algorithm (see [15] for more detail) takes the database D of all saved sequences of events as input. Here, a sequence of events is recorded for each execution of CTS. An event X=(i_1, i_2, … i_n) contains a set of items i_1, i_2, … i_n, and represents one cognitive cycle. For each event, (1) an item represents the coalition of information-codelets that was broadcasted during the cognitive cycle, (2) an optional four items having numeric values indicates the four emotional valences (high threat, medium fear, low threat, compassion) that are associated with the broadcasted coalition, as explained in section 3, and (3) an optional item represents the executed behavior, if one was executed during that cycle. Formally, an events sequence is denoted s = <(t_1,X_1), (t_2,X_2),…, (t_n,X_n)>, where each event X_k is annotated with a timestamp t_k indicating the cognitive cycle number. The algorithm extracts partial or complete sequences occurring in the database more than a minimal number of times defined by the user (*minsup*).

Table 1. A Data Set of 6 Sequences

ID	Events sequences
S1	<(0, c1 e1{0.8}), (1, c2 e2{0.3} b1), (2, c4 b5)>
S2	<(0, c1 e1{0.8}), (1, c3), (2, c4 b4), (3, c5 b3)>
S3	<(0, c2 e2{0.3}), (1, c3), (2, c4), (3, c5 b3)>
S4	<(0, c3), (1, c1 e1{0.6} b4),(2, c3)>
S5	<(0, c4 b4), (1, c5), (2, c6)>
S6	<(1, c1 e1{0.6} b4), (2, c4 b4), (3, c5)>

Table 1 shows an example of a database produced by user manipulation of canadaarm2 in the virtual world. We chose 2 short sequences in this example. The first event of sequence S1 shows that during cognitive cycle 0, due to arm manipulation coalition c1 was broadcasted and that an emotional valence of 0.8 for emotion e1 (high threat) was associated with the broadcast. The second event of S1 indicates that at cognitive cycle 1, coalition c2 was broadcasted with emotional valence 0.3 for emotion e2 (medium fear) and that behavior b1 was executed. Table 2 shows some sequences obtained (see [15] for more details) from the application of the algorithm on the database of Table 1 with a *minsup* of 32 % (2 sequences) and no time constraints. The first frequent pattern is <(0, c1 e1 {0.7}), (2, c4)>, which was found in sequences S1, S2, S4 and S6. Because the events containing e1 in these sequences have numeric values 0.8, 0.8, 0.6 and 0.6, the algorithm calculated the average when extracting that pattern, which resulted in the first event having e1 with value {0.7}. Because this pattern has a support of 66 % (4 out of 6 sequences), which is higher than minsup, it is deemed frequent.

Table 2. Example of Events Sequences Extracted

Mined sequences	Support
<(0, c1 e1{0.7}), (2, c4)>	66 %
<(0, c3), (2, c5 b3)>	33 %
<(0, c4 b4), (1, c5)>	50 %
<(1, c3), (2, c4), (3, c5 b3)>	33 %
…	…

4.3 The Episodic Learning Process

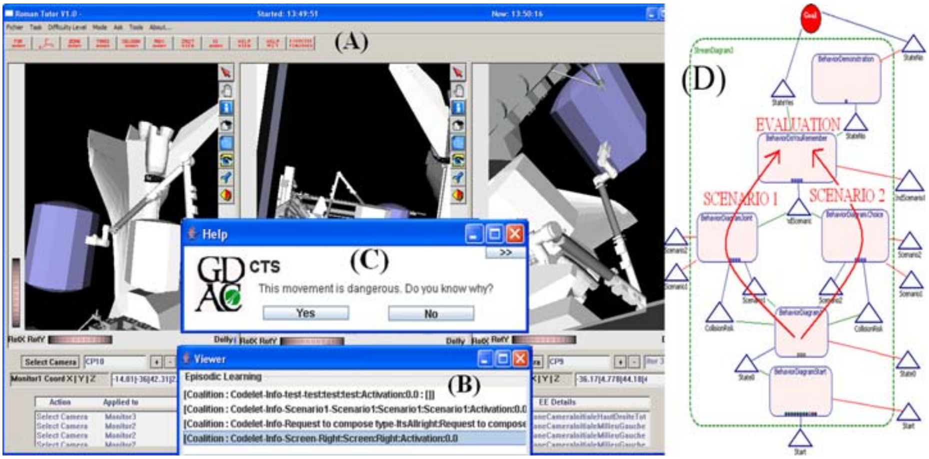
The episodic learning mechanism constantly adapts to the learner or other agents behavior by intervening in the coalitions selection phase of CTS, as follows. At each cognitive cycle, before choosing the candidate coalition to be broadcasted, all the frequent events sequences are scanned for finding those that matches with the coalitions broadcasted at previous cognitive cycles. The last n broadcasted coalitions are represented as a sequence Sc=<(0,c_1), (1,c_2)... (n,c_n)>. The "*episodic learning*" mechanism checks all frequent sequences to find each sequence Sa=<(t_1,a_1), (t_2, a_2), ... (t_n,a_n)> such that for a number k > 1, the sequence defined by the last k broadcasts of Sc <(t_{n-k+1}, c_{n-k+1}), (t_{n-k+2},c_{n-k+2}) ... (t_n,c_n)> is included in Sa and there is an event following that sequence occurrence that contains a coalition that is candidate for being broadcast in the current cognitive cycle. The "episodic learning mechanism" then selects among these sequences the one that is the most probable of being emotionally positive. To do this, it first computes for each sequence the sum of emotional valences associated to each broadcast occurrence in the sequence. For each sequence having a positive or negative sum, it multiplies this sum with the sequence support to obtain the sequence strength. The "episodic learning mechanism" selects the sequence with the most strength. If no sequence has a positive strength, it selects the one with a sum of zero that is the most frequent, or the sequence that has the least negative strength, in that order. The broadcast in event a_{k+1} for the largest k will be broadcast, in favor of other coalitions waiting to be broadcasted.

5 Experimentation

We performed various experiments to validate this new Episodic learning mechanism. A total of 20 users (a mix of novices, intermediates and experts) were invited to execute various experiments using the CanadarmII robot simulator with integrated CTS. To begin this, we needed to establish a gross model of classification for users. We determined classification from highest to lowest performance user groups using previous data and data mining algorithms: (1) the high-performance group showing preferential focus for the relevant[6] paths during arm manipulation; (2) the medium-performance group lacking focus and exploring devious paths; (3) the low-performance group having no focus and adopting random paths indicating thus complete unawareness of the virtual world's conceptual structure. To explain how CTS evaluates and interact with users, one experiment conducted with user 3 is here detailed. After some preliminary interactions, CTS classified him as a low-performance group user. User 3 tended to make frequent mistakes when he was asked to guess the arm distance from a specific part of the ISS and its name (Figure 2.A). Obviously, this situation caused collision risks between the arm and ISS and was thus a very dangerous situation. This situation was implemented in the CTS's Behavior Network (Figure2.D).

In this situation, the CTS Episodic Learning mechanism (Figure 2.B), collaborating with Pedagogical knowledge mechanism, had to make a decision between giving a

[6] As determined by an expert.



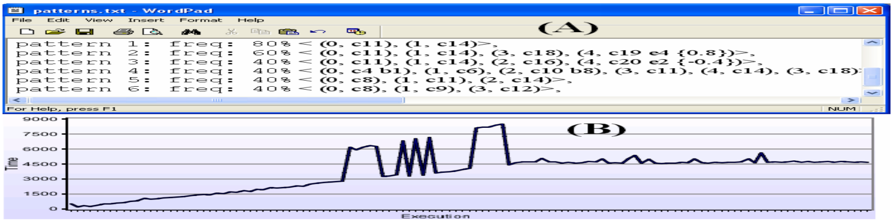
This movement is dangerous. Do you know why?

Episodic Learning

```
[Coalition : Codelet-Info-test-test:test:test:Activation:0.0 : []]
[Coalition : Codelet-Info-Scenario1-Scenario1:Scenario1:Scenario1:Activation:0.0
[Coalition : Codelet-Info-Request to compose type-ItsAllright:Request to compose
[Coalition : Codelet-Info-Screen-Right:ScreenRight:Activation:0.0
```

Fig. 2. (A) Simulator interface (B) Episodic Learning Viewer (C) CTS Intervention (D) Part of the CTS Behavior Network

direct solution "*You should move joint SP*" (Figure 2.D Scenario 1) or a brief hint such as "*This movement is dangerous. Do you know why*?" (Figure2.D Scenario 2). As mentioned above, CTS classified and learned about the user's profile interacting with different users. With user 3, the experiments revealed that it is better to first give a hint to the user and then ask him to think about the situation before giving him the answer. At the end of each scenario, it is also best that CTS ask an evaluation question to verify the effectiveness of its interventions (Figure 3. *Evaluation*). As we mentioned previously, two paths exist for CTS Behavior Network to solve this problem (Figure 2.D). In the case of User 3, CTS chose the second option and presented a hint followed by an answer (Figure2.C). User 3 did not remember the answer, so a help (a hint and not a solution) was proposed to him to push him to think more about the situation. He still did not remember, so CTS finally signaled to User 3 that this situation would end with a collision with the ISS. During the interaction with this user, CTS learned the valences of each trace (Figure 2.D Scenario1 and 2) with its nodes and all emotional valences were saved at the end of each execution. The average length of the stored sequences was of 22 events. One partial trace saved when CTS gave a hint (scenario 2) to User 3 was <(13, c11), (14, c14), (15, c15), (16, c18), (17, c19 e4 {0.8}>. In this trace, the positive valence 0.8 for emotion e4 (compassion) was recorded because the learner answered to the Evaluation question correctly after receiving the hint. Thus during CTS's interaction with User 3, when User 3 gave the correct answer to CTS, the EM gave more positive valences to the scenario which might have been chosen in the future by EPL mechanism. In another partial trace saved by CTS, when User 3, having received some help, answered incorrectly to a question (scenario 1) <(16, c11), (17, c14), (18, c16), (19, c17), (20, c20 e2 {-0.4}>, CTS associated the negative valence -0.4 to emotion e2 (medium fear). After five executions, the memory consolidation phase extracted ten frequent event sequences, with a minimum support (*minsup*) higher than 0.25. This way, had the CTS have had to face the same problem in the future, it might have chosen between scenario 1 and



```
pattern 1: freq: 80% < (0, c11), (1, c14)>,
pattern 2: freq: 60% < (0, c11), (1, c14), (3, c18), (4, c19 e4 {0.8})>,
pattern 3: freq: 40% < (0, c11), (1, c14), (2, c16), (4, c20 e2 {-0.4})>,
pattern 4: freq: 40% < (0, c4 b1), (1, c6), (2, c10 b8), (3, c11), (4, c14), (3, c18)
pattern 5: freq: 40% < (0, c8), (1, c11), (2, c14)>,
pattern 6: freq: 40% < (0, c8), (1, c9), (3, c12)>,
```

Fig. 3. (A) Part of the frequent sequential patterns (B) Data mining algorithm performance for each execution

2. The episodic learning process evaluates all consolidated patterns and tries to detect the best pattern for each situation having ended by self-satisfaction (OCC model) in CTS. CTS chooses the pattern < (0, c11), (1, c14), (3, c18), (4, c19 e4 {0.8})>, because it contains the most positive emotional valence, has the highest frequency, and many of the events <(0, c11), (1, c14)> match with the latest events executed in CTS. In our model, CTS will thus choose the path which gives it greatest self-satisfaction (Figure 2.D scenario 2). Concretely, this was achieved by broadcasting coalition c18 (Scenario 2) instead of coalition c16 (Scenario 1). If the emotional valence is not as positive as was the case with User 3, CTS can choose scenario 1 rather than scenario 2. Moreover, because the set of patterns is regenerated after each CTS cognitive cycle, some new patterns are created and others disappear. This may cause some changes in CTS's behavior. This will guarantees that CTS's architecture adapt better to any situation and dynamically learn how to associate an episode to one or more scenarios in the BN. Lastly, we conducted a preliminary experiment to evaluate the capacity of the data mining algorithm (see figure 3). We generated up to 122 executions choosing randomly between Scenario 1, 2 (Figure 3.B) shows extracted patterns after each CTS execution. The graph showed that the time increase linearly after each CTS execution. As shown in Figure 3.B, the algorithm showed a stable behavior after 70 executions. It also ensures that it learns how to choose the path that is the most probable of bringing self-satisfaction. We were also able to extract the frequent sequential patterns. As regards to this matter, the results remained inconclusive. In our preliminary study, the algorithm did not always behave in a constant linear fashion. This led to an increase in reaction time and pattern extraction time.

6 Conclusion

As far as we know, no cognitive agent presently uses data mining algorithms to improve their behavior. Nonetheless, they have proven very useful to extract significant information from the huge amount data that they have to handle. The interaction between an agent and its dynamic environment also generates large amounts of data. The episodic learning algorithm used in this work is inspired from a memory consolidation theory which is biologically plausible. The collaboration between the Emotional Mechanism and this Episodic Learning helps to choose the behaviors that are most likely to bring the agent to a self-satisfactory emotional state. In this work, we have tested the new CTS architecture to verify its capability to adapt to users. We are currently working on performing more experiments with huge amount of data to evaluate the data mining algorithm used in this experiment. With these experiments, we should perfect our approach. We also plan to integrate CTS in other tutoring systems.

Acknowledgement

Our special thanks to the GDAC's members for their collaborations and the Fonds Québécois de la Recherche sur la Nature et les Technologies and the Natural Sciences and Engineering Research Council for their financial support.

References

1. Tulving, E.: Precis of Elements of Episodic Memory. Behav. and Brain Sciences (1984)
2. Dolan, R.J., Lane, R., Chua, P., Fletcher, P.: Dissociable Temporal Lobe Activations during Emotional Episodic Memory Retrieval. Neuroimage 11, 203–209 (2000)
3. Bower, G.H.: How might emotions affect learning. In: The Handbook of Emotion and Memory: Research and Theory. Lawrence Erlbaum, Hilsdale (1992)
4. Dubois, D., Poirier, P., Nkambou, R.: What Does Consciousness Bring to CTS? LNCS (LNAI), vol. 5091, pp. 803–806. Springer, Berlin (2007)
5. Purves, D., Brannon, E., Cabeza, R., Huettel, S.A., Labar, K.S., Platt, M.L., Woldorff, M.G.: The principles of cognitive neuroscience. E. Sinauer Associates (2008)
6. Faghihi, U., Poirier, P., Dubois, D., Gaha, M., Nkambou, R.: Implementation of Emotional Learning for Cognitive Tutoring Agents. In: Gelbukh, A., Morales, E.F. (eds.) MICAI 2008. LNCS, vol. 5317. Springer, Heidelberg (2008)
7. Ortony, A., Clore, G., Collins, A.: The cognitive structure of emotions. Cambridge University Press, Cambridge (1988)
8. Agrawal, R., Srikant, R.: Mining Sequential Patterns. In: Proceedings of the 11th Int'l. Conference on Data Engineering, March 1995, pp. 3–14 (1995)
9. Anderson, J.R.: Rules of the mind. Erlbaum, Mahwah (1993)
10. Sun, R.: The CLARION cognitive architecture: Extending cognitive modeling to social simulation Cognition and Multi-Agent interaction. Cambridge University Press, Cambridge (2006)
11. Franklin, S., Patterson, F.G.J.: The LIDA architecture: adding new modes of learning to an intelligent, autonomous, software agent. Integrated Design and Process Technology. Society for Design and Process Science (2006)
12. Baars, B.J.: In the Theater of Consciousness. Oxford Univ. Press, Oxford (1997)
13. Hofstadter, D.R., Mitchell, M.: The Copycat Project: A model of mental fluidity and analogy-making. In: Holyoak, K.J., Barnden, J.A. (eds.) Advances in connectionist and neural computation theory, vol. 2. Ablex, Norwood (1994)
14. Maes, P.: How to do the right thing. Connection Science (1989)
15. Fournier-Viger, P., Nkambou, R., Mephu Nguifo, E.: A Knowledge Discovery Framework for Learning Task Models from User Interactions in Intelligent Tutoring Systems. In: Gelbukh, A., Morales, E.F. (eds.) MICAI 2008. LNCS, vol. 5317, pp. 765–778. Springer, Heidelberg (2008)
16. Nkambou, R., Belghith, K., Kabanza, F., Khan, M.: Supporting Training on Canadarm Simulator using a Flexible Path Planner. In: Proc. of AIED 2005 (2005)
17. Najjar, M., Fournier-Viger, P., Mayers, A., Bouchard, F.: Memorising remembrances in computational modelling of interrupted activities. In: Proc. of CINC 2005 (2005)
18. Faghihi, U., Dubois, D., Nkambou, R.: Human-Like Learning Methods for a "Conscious" Agent. In: Advanced Intelligent Computing Theories and Applications (ICIC 2007) (2007)
19. Sidney, D.M., Franklin, S., Uma, R., Bernard, B.: A cognitive science based machine learning architecture (2006)

Multi-source Signals Guiding Swarm Robots Search*

Songdong Xue[1,2], Jianchao Zeng[2], Jing Du[3], and Xiaomei Yang[2]

[1] College of Electrical and Information Engineering,
Lanzhou University of Technology,
85 Langongping, Lanzhou, Gansu 730050, China
[2] Complex System and Computational Intelligence Laboratory,
Taiyuan University of Science and Technology,
66 Waliu Road, Taiyuan, Shanxi 030024, China
[3] North Automatic Control Technology Institute,
China North Industries Group Corporation,
351 South Sports Road, Taiyuan, Shanxi 030006, China
xuesongdong@gmail.com, zengjianchao@263.net
http://www1.tyust.edu.cn/yuanxi/yjjg/index.htm

Abstract. Victim search with swarm robots on gas outburst spot in coal mines is investigated. Upon modeling swarm robotic system with extended particle swarm optimization, heterogeneous signals are fused by each robot independently as fitness evaluate. To fuse signals of intermittent sound of call for help, periodic radio frequency waves and continuous gas, concepts of detection-success binary logic and perceptual event are introduced into describing detection process. Based on statistical properties, detectable range, localization type and accuracy, priority is determined with information entropy-based criteria. Then the best-found position is decided in a distributed fashion through election operation. Simulation conducted in closed signal propagation environment indicates the approximate relation between fusion value and distance, i.e., the nearer the robot is far away from target, the higher the fusion of signals.

Keywords: Swarm intelligence, swarm robotics, target search, particle swarm optimization, signals fusion.

1 Introduction

When working miners are confronted with gas outburst even gas explosion accidents in closed roadways, they would be likely to lose touch with outside. Unfortunately, search operations tend to be difficult due to the extreme risk. Swarm robots may therefore be used to carry out such missions taking the place of

* This work was supported in part by National Natural Science Foundation of China under Grant No. 60674104 and by Shanxi Science Fund for Young Scholars under Grant No. 2006021019.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 556–565, 2009.

human beings. Extending particle swarm optimization (PSO) to model swarm robots, Marques et al. [1], Pugh et al. [2], and Hereford et al. [3] investigate source/target search respectively. The key of their approaches is to detect target signals as fitness evaluate in PSO so as to decide the best-found position by "fitness" comparison, since PSO-style search working depends on guide by the best experience. However, robot there is assumed to be equipped with a single sensor to measure external signals, which has theoretical significance only. In fact, there are multiple kinds of heterogeneous signals, including intermittent sound of call for help and periodic RF waves as well as continuous gas on disaster spot. It is need to decide the best positions of both individual robot itself and the best of population within its neighborhood under such conditions.

The remainder of this paper proceeds as follows: In Section 2 the properties of signals propagating in search space are introduced. Then a fusion framework is presented in Section 3. To examine the validity of fusion approach, simulations are conducted in Section 4. Finally, we conclude in Section 5.

2 Signals Emitted by Target

With mathematical models of signals propagation, we can generate a set of theoretically computed signal strength data akin to the empirical data set to study control strategy rather than collect the data on spot.

Sound of Call for Help. The identical model may be used for both propagation and detection as to the same space. Compared with the size of environment, the mouth of victim can be considered as a point sound source. Let N robots equipped with acoustic sensors construct a mobile sensor field, where an immovable target emits omnidirectional acoustic signals. The signal energy measured on the i^{th} sensor over time interval t, denoted by [4]:

$$y_i(t) = g_i \frac{s(t - t_i)}{|r(t - t_i) - r_i|^{\alpha}} + \varepsilon_i(t) \tag{1}$$

where t_i is time delay for sound propagates from target to the i^{th} robot, $s(t)$ is a scalar denoting energy emitted during sampling time t; $r(t)$ coordinates of target during t; r_i coordinates of the i^{th} stationary sensor; g_i gain factor of the i^{th} acoustic sensor; $\alpha(\approx 2)$ energy decay factor, and $\varepsilon_i(t)$ cumulative effects of modeling error of g_i, r_i, α and the additive observation noise of $y_i(t)$, see Fig. 1.

RF Waves. Typically, underground mine personnel tracking systems work on the basis of radio frequency identification (RFID) technologies today. Such a system has basic components including readers and tags. The latter is categorized as either passive or active [5]. As for coal mine application, a tag is often mounted on a miner's helmet with his lamp [6]. For a radio channel, the transmitted signal reaches receiver via multiple paths [7]

$$P(d) = P(d_0) - 10\alpha \lg \frac{d}{d_0} - \xi \tag{2}$$

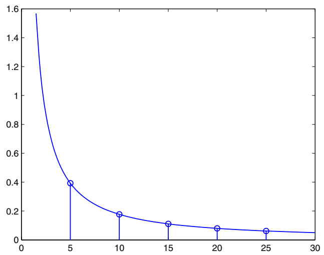


Fig. 1. Acoustic energy loss fitting

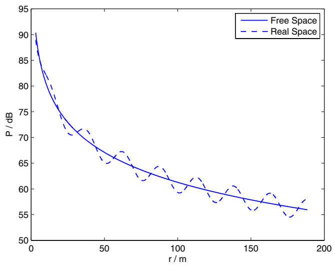


Fig. 2. Log-normal signal loss distribution

where α indicates loss rate [8], $P(d_0)$ is signal power at reference distance d_0 [9] and d transmitter-receiver distance. The value of $P(d_0)$ can be derived empirically or obtained from the wireless network hardware specifications. In general, ξ value is derived empirically, see Fig. 2.

Toxic Gases from Gas Outburst. Gas in coal mines will diffuse quickly in closed roadways after gas outburst. The pervasion process can be described as affecting by some odor point sources. For convenience, they can be viewed as only one by linear combination. Let the projection of leak point on the ground be origin, average direction of downwind x-axis, a right hand 3-D reference frame can be set. Then, we can calculate gas concentration in any point on the ground ($z = 0$) following the law [1,10,11]:

$$C(x, y, t) = \frac{Q}{2\pi\sigma_y(x,t)\sigma_z(x)} \exp\{\frac{(y(t) - y_0(x,t))^2}{-2\sigma_y^2(x,t)}\} \tag{3}$$

where Q represents release rate, plume center y_0, width w and height h as a function of time t and downwind distance x. $\sigma_y(x,t) = w(x,t)/\sqrt{2\pi}$, $\sigma_z(x) = h(x)/\sqrt{2\pi}$. Fig. 3 shows an example of a time averaged Gaussian plume [1,11,12]. Further, heterogeneous signals distribution in search environment can be shown in Fig. 4.

Signals Propagation Environment. The space is divided into six sub-areas (numbering Area 1–6) according to the distribution characteristics of signals. The lines in Fig. 4 represent the minimum detectable signal contours corresponding to thresholds 0.0016 kg/m^3 and −90 dBm and to maximum detectable ranges 200 m and 45 m respectively [4,5,8,11]. It is need to point out that threshold of sound is not given definitely because it is closely related to the sensor sensitivity. Hence, given a specific power (milliwatt magnitude) of call for help in a loud voice, our attention lines in finding how far the emitted sound signals can reach. In the part of simulation, we make an experiential but reasonable assumption.

3 Fusion

Gas diffusion distance may be up to several hundreds of meters [1,10,11]; the detectable range of RF waves with frequency $f = 7.5$ s emitted by active tags can

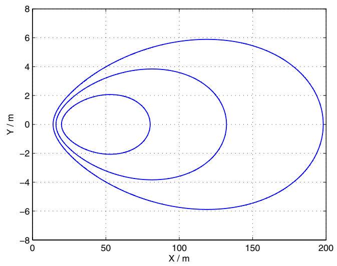


Fig. 3. Gas distribution on the ground

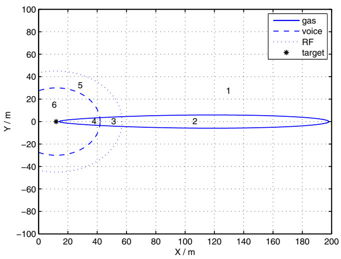


Fig. 4. Signals distribution in search space

be 150 ft (≈ 45 m) [5] and the localization accuracy with RF RSSI method can be 2 m [13]; the detectable range of sound usually reaches not more than 30 m [4], and the estimation error with sound RSSI method can be 50% [14]. Therefore, detectable range, statistical properties, localization types and accuracy have to be considered simultaneously.

3.1 Perception Event

Introduce a 2-value logic into describing perceptual process now so that we can define perception event in advance. Let A_i ($i = \mathrm{GAS}, \mathrm{RF}, \mathrm{CALL}$) be perceptual event in event space $\Omega = \{0, 1\}$. Then $A_i = 1$ represents detect-success (beyond threshold) and $A_i = 0$ detect-failure. At the same time, we normalize those measurement readings beyond threshold to $\mathrm{Nm}_i \in (0, 1)$. Clearly, events A_i and A_j ($i \neq j$) are mutually independent and the probability of each event can be calculated with its statistical properties. Further, let $(A_{i\mathrm{GAS}}(t), A_{i\mathrm{RF}}(t), A_{i\mathrm{CALL}}(t))$ be the joint perception event of robot i at time t, then there may be $2^3 = 8$ joint events according to the time characteristic of signals distribution. Again, considering the spatial distribution feature of signals, it is necessary to simplify the complexity of temporal-spatial distribution of heterogeneous signals. Those robots in signal blind area (Area 1) attempt to capture signal clues independently in a spiral move manner to further search for target locally [1,15], without directed by swarm intelligence principle locally, while the ones in Area 2–6 do so [1]. Thus, it is easy to know that there are six joint events everywhere except source. The possible joint events occurred in each sub-area are listed in Table 1. The denoting of these joint events can be encoded with three-bit binary number. Since the source is characteristic of such encode, we can also express it with 3 × 1 “characteristic” row vector **C**. Finally, take the above Nm_is of measurement readings to replace the corresponding elements “1” in vector **V**.

3.2 Virtual Communication

We view signals measuring as continuous communication between robots and target. Each robot possesses its own channel, with source (target) and destination (robot). The individual robot discretizes three types of measurement readings

Table 1. Joint perception events $(A_{\text{GAS}}, A_{\text{RF}}, A_{\text{CALL}})$ in sub-areas of search space

Source Encode	Sub-Area	Possible Event	"Received" Encode
111	1	$(0,0,0)$	000
111	2	$(1,0,0)$	100
111	3	$(1,0,0)$, $(1,1,0)$	100, 101
111	4	$(1,0,0)$, $(1,1,0)$, $(1,0,1)$, $(1,1,1)$	100, 110, 101, 111
111	5	$(0,0,0)$, $(0,1,0)$	000, 010
111	6	$(0,0,0)$, $(0,0,1)$, $(0,1,0)$, $(0,1,1)$	000, 001, 010, 011

to 1-bit binary digit respectively with corresponding threshold. As for the same signal emitted from source, robots in different sub-areas may obtain different results due to the effect of distance. Consequently, while there are $3^2 = 8$ encodes of full permutation in source, the "received" encodes by robots in different sub-areas may vary. We suppose the duration of emission by target can guarantee the detection success in sampling period (400 ms here). But the transmission time from source to destination is small enough so as to be ignored. Thus, the detection process can transfer continuous information to time- and amplitude-discrete random signal series. Since we suppose Δt be sufficient small interval, the above perception event occurs once at most every sampling process.

3.3 Information Entropy

We can calculate the information entropy from the received information encodes through virtual communication process.

Toxic Gases from Gas Outburst. Robots start to locally search for target as soon as gas can be sensed in global search stage [1]. As to the continuous gas diffusion, event $A_{\text{GAS}} = 1$ will occur at any time t in Area 2–4 [1,15]. Then $P\{A_{\text{GAS}}(t)\} = 1$, i.e., this information is decisive, say, $H(X_{\text{GAS}}) = 0$.

RFID Waves. The perception process of RF signals $\{X_{\text{RF}}(t), t \geq 0\}$ can be viewed as Poisson process with intensity λ_{RF}. As the process has stationary independent increment, those equal intervals may lead to equal probability of $A_{\text{RF}} = 1$. Suppose the sampling period Δt is sufficient small time interval and it satisfies the sampling theory. The event $A_{\text{RF}} = 1$ occurs once at most every sampling process. In addition, taking the assumption of event completion and virtual communication into account, events $A_{\text{RF}} = 1$ and $A_{\text{RF}} = 0$ are contrary ones, then we can draw a conclusion $P\{A_{\text{RF}}(t) = 0\} = 1 - P\{A_{\text{RF}}(t) = 1\}$. And we can further capture the relation by computing their event probabilities

$$P\{A_{\text{RF}}(t) = 1\} = P\{X_{\text{RF}}(t + \Delta t) - X_{\text{RF}}(t) = 1\} = e^{-\lambda_{\text{RF}} \Delta t} \lambda_{\text{RF}} \Delta t \qquad (4)$$

$$P\{A_{\mathrm{RF}}(t)=0\}=P\{X_{\mathrm{RF}}(t+\Delta t)-X_{\mathrm{RF}}(t)=0\}=e^{-\lambda_{\mathrm{RF}}\Delta t} \tag{5}$$

$$e^{\lambda_{\mathrm{RF}}\Delta t}=\lambda_{\mathrm{RF}}\Delta t+1 \tag{6}$$

Accordingly, entropy of RF signals in information source at t can be calculated

$$H(X_{\mathrm{RF}})=\lambda_{\mathrm{RF}}\Delta t+(e^{-\lambda_{\mathrm{RF}}\Delta t}-1)\log(\lambda_{\mathrm{RF}}\Delta t) \tag{7}$$

Sound of Call for Help. The detected sound of call for help $\{X_{\mathrm{C}}(t), t\geq 0\}$ be Poisson process with intensity λ_{C}. We can further determine the probability detect-success and detect-failure of call for help, as well as the entropy of such sound at any time t, as shown below

$$H(X_{\mathrm{C}})=\lambda_{\mathrm{C}}\Delta t+(e^{-\lambda_{\mathrm{C}}\Delta t}-1)\log(\lambda_{\mathrm{C}}\Delta t) \tag{8}$$

3.4 Weight

A criterion of sensor weight can be used by individual robot to signals fusion. Among all factors, apart from information entropy, the localization type and accuracy with RSSI method should also be considered [16]. For instance, gas source is not same as the target location, i.e., we localize the target indirectly by localizing gas source based on the fact that one should move quickly upwind in risk avoiding poison gas leakage. Consequently, such estimate may get the worst accuracy (200 m assumed). While the RSSI-based estimate with RF or sound intensity can localize target directly. But the two types of signals have different accuracy in location estimate, see Table 2 for details [1,4,5,13,14]. The criterion is used to determine the contribution to fusion by sensor. As shown in (9), the entropy, localization type and accuracy are all required to be integrated by weighted sums, which can be viewed as the priority w of sensor.

$$w_i=\frac{aH(X_i)}{\sum\limits_i H(X_i)}+b\kappa+\frac{c}{\tau_i\sum\limits_i\frac{1}{\tau_i}}, i=\mathrm{GAS},\mathrm{RF},\mathrm{CALL} \tag{9}$$

where logic variable κ represents localization type, “indirect” is assigned 0 and “direct” 1. τ is localization accuracy and $a,b,c\in(0,1]$ are all positive coefficients need to be determined empirically. Then, take three weight values as elements to construct a 1×3 column vector $\mathbf{W}$.

3.5 Signals Fusion

An fusion mechanism for making decision on the best positions is discussed here, being suitable for deciding on cognitive of individual and social of swarm. The mechanism can be expressed with weighted sums operation using vectors $\mathbf{V}$ and $\mathbf{W}$, i.e., obtaining the fusion by calculating the inner product of two vectors $fusion=\mathbf{V}_{(1\times 3)}\cdot\mathbf{W}_{(3\times 1)}$.

Table 2. Characteristics of perceptual signals propagated in search space

Signal	Entropy	Detectable Range(m)	Localization Type	Accuracy (m)
GAS	0	200	indirect	200
RF	0.0156	45	direct	2
CALL	0.2055	30	direct	15

3.6 Description of Fusion Algorithm

We can describe the full-distributed fusion algorithm run on individual robots, taken as an embedded module of control system [16]. First of all, we assume each robot has an unique ID, carrying a set of sensors and a on-board fusion module so as to measure surrounding environment and fuse the detected signals independently. Besides, all sensors can react to signals in sufficient short time. Third, we construct a character structure with respect to target and denote it as "ID"+"Position"+"$fusion$".

```
– Confirm ID and current position $iPos$;
– Initialize
    • Set counter $t \leftarrow 0$;
    • Set $(A_{i\mathrm{GAS}}, A_{i\mathrm{RF}}, A_{i\mathrm{CALL}})_{t=0} = 000$;
    • Set $fusion = 0$;
    • Construct "ID"+"Position"+"$fusion$";
    • Set best position of itself $ibPos \leftarrow iPos$ ;
    • Set best position of neighborhood $sbPos \leftarrow iPos$;
– While termination is not triggered
    • Detect signals emitted by target;
    • Discretize to 0 or 1 by comparing measurements with thresholds;
    • Format data with characteristic structure;
    • If $(A_{i\mathrm{GAS}}, A_{i\mathrm{RF}}, A_{i\mathrm{CALL}})_t = 000$, then keep silence;
    • Otherwise
        ∗ Elect $ibPos$ and update;
        ∗ Broadcast data within its neighborhood;
    • Listen for others, if receive data containing $(A_{j\mathrm{GAS}}, A_{j\mathrm{RF}}, A_{j\mathrm{CALL}}) \neq$
      000, then elect $sbPos$ and update;
    • $t \leftarrow t + 1$;
– End.
```

4 Simulation and Discussion

To elaborate how to fuse the specific heterogeneous signals and how to decide the best positions, the simulations are designed and conducted for the purpose. First, virtual signal generators are arranged where same as target situates, emitting

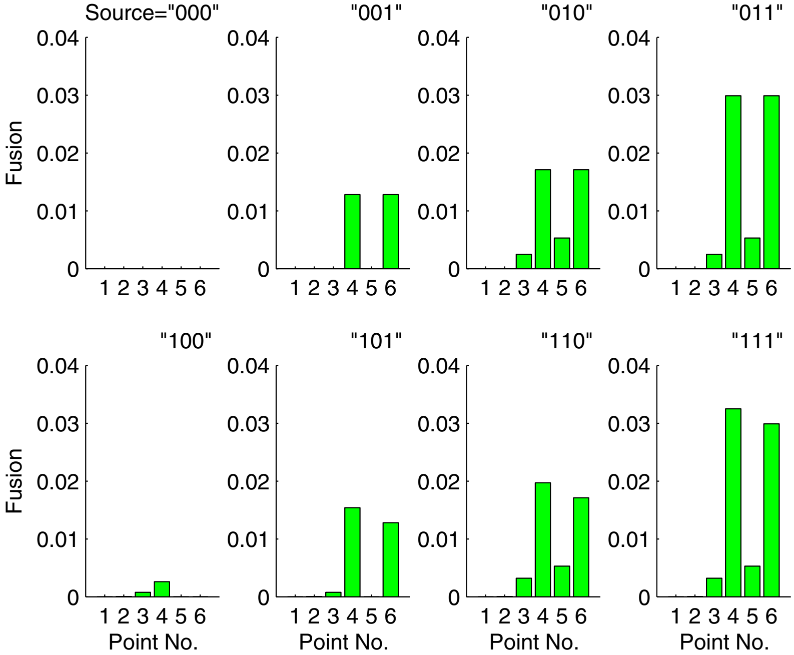


Fig. 5. Fusion results at the arranged six measuring points under different encodes of information source. Note that the title Source="000" of the left corner sub-figure represents no GAS, no RF, and no CALL signals are emitted when sampling. One can understand the other cases by analogy. Besides, the fusion values have not any dimension.

signals following their own time characteristic. Then, a series of detection points are set in signal Area 1–6. Our task is to investigate what happened in each information sink (robot) when different combination of signals is emitted from source by virtually measuring and fusing. We observe for sufficient long time until all eight encodes transmitted from source. Then we try to find the relation between distance and fusion result.

4.1 Signals Generating

Consider the properties of a given Poisson process with intensity λ. The successive coming time of events obey exponential distribution with mean $\frac{1}{\lambda}$. We can empirically set the value in some interval, for example, the upper bound and lower bound can set to 0.01 and 0.001 respectively, i.e., $\lambda_C \in (0.001, 0.01)$, while the intensity of RF signals can be $\lambda_{RF} = 0.1333$ according to its primitive definition, which reflect the temporal characteristics of target signals.

4.2 Arrangement of Measuring Point

We set a series of measuring points, assigning one with each sub-area. Different points are different far away from the source. Note that a pair of points in different areas having the same distance value are arranged to study the relation between fusions at the same time.

4.3 Main Parameter Settings

We simulate fusion algorithm using parameter configuration $a = 1, b = 0.001, c = 1, \lambda_{\mathrm{C}} = 0.2055, \lambda_{\mathrm{RF}} = 0.0156$. For convenience, target is fixed to $(0, 0)$ all time and the coordinates of six measuring points are $(100, 40)$, $(150, 0)$, $(40, 0)$, $(20, 0)$, $(0, 35)$, $(0, 20)$ orderly. Meanwhile, we focus on if the coverage of all joint events occur in sufficient long time rather than the moments.

4.4 Results and Discussions

The fused values in simulation are shown in Fig. 5, from which robots can "find" the best positions by simple election operation. It is easy to find that

- The bigger the fusion value, the nearer the measuring point from target;
- As for No. 4 and No. 6 point, the fusion results are the same in some cases of $Source = 001, 010, 011$, and different in some cases of $Source = 101, 110, 111$ although they are equal to distance of target. We may explain it in this manner: robots searching for target depend on measurements because they do not know the position of target. While the two points are located in different sub-areas, the situation of signals cover is different.

5 Conclusions

As for PSO-based control of swarm robots, the experience both of individual robots and of population is required. In order to decide the best positions, we take the characteristic information of target, such as intensity or concentration of different signals emitted by target, as the "fitness". Therefore, the problem of multi-source signals fusion is proposed. To this end, we model the process of signals measurement with robot sensors as virtual communication. Then, the detected target signals can be viewed as transmitted encodes with respect to information source. We thereupon present some concepts of binary logic and perceptual event to describe the "communication“ between target and robots. Besides, we also put forward information entropy-based fusion criteria and priority to fuse signals and election mechanism to decide the best positions on the basis of space-time distribution properties of target and robots. Simulation conducted in closed signal propagation environment indicates the approximate relation between fusion and distance, i.e., the nearer the robot is far away from target, the higher the fusion of signals. But, we must point out that the proposed fusion method has mathematical rather than physical significance. The numerical relations between fusion and distance merit more work in the future.

References

1. Marques, L., Nunes, U., De Almeida, A.T.: Particle Swarm-Based Olfactory Guided Search. Autonomous Robotics 20, 277–287 (2006)
2. Pugh, J., Martinoli, A.: Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization. In: Proc. of the 4th IEEE Swarm Int'l. Symposium, Honolulu, HI, USA (2007)
3. Hereford, J., Siebold, M.: Multi-robot search using a physically-embedded Particle Swarm Optimization. Int'l. J. of Computational Intelligence Research 4(2), 197–209 (2008)
4. Li, D., Hu, Y.U.: Energy-Based Collaborative Source Localization Using Acoustic Microsensor Array. EURASIP J. on Appl. Sig. Processing 4, 321–337 (2003)
5. Ni, L.M., Liu, Y.H.: LANDMARC: Indoor Location Sensing Using Active RFID. Wireless Networks 10, 701–710 (2004)
6. You, Z.Q., Li, S.J., et al.: Theory and Applications of RFID. PHEI, Beijing (2004) (in Chinese)
7. Bahl, P., Padmanabhan, V.N.: RADAR: An In-Building RF-based User Location and Tracking System. In: Proc. of IEEE INFOCOM, vol. 3, pp. 775–784 (2000)
8. Feng, H.R., Yuan, R.X., Mu, C.D.: Comparison of RSS-Based Localization Algorithms for Wireless Networks. Eng. & Appl. of Comp. 32, 1–3,11 (2006) (in Chinese)
9. Shi, P., Xu, F.Y., Wang, Z.X.: A Maximum-Likelihood Indoor Location Algorithm Based on Indoor Propagation Loss Model. Sig. Processing 21(5), 502–504,474 (2005) (in Chinese)
10. Jia, Z.W., Jing, G.X., Zhang, Q.: Analysis on the Diffusion of Toxic Gases from Gas Explosion and the Determination of Risk Area. China Safety Science Journal 17(1), 91–95 (2007) (in Chinese)
11. Chong, X.H., Ren, Z.G., Ding, X.G.: Interfacial and Numerical Simulation of Poison Gas Diffusion. Sciencepaper Online (2007), `http://www.paper.edu.cn/downloadpaper.php?serial_number=200709-38&type=1` (in Chinese)
12. Zhou, B., Zhang, G.S.: Numerical Simulation for the Leakage Diffusion of Harmful Materials. Industrial Safety and Environment Protection 31(10), 42–44 (2005) (in Chinese)
13. Sun, P.G., Zhao, H., et al.: Research on RSSI-based Location in Smart Space. Acta Electronica Sinica 35(7), 1240–1245 (2007) (in Chinese)
14. Deng, H.B., Jia, Y.D., et al.: A Planetary Rover Localization Algorithm Based on Wireless Sensor Network. Robot. 29(4), 384–388 (2007) (in Chinese)
15. Hayes, A.T.: Self-Organized Robotic System Design and Autonomous Odor Localization. California Inst. of Tech., CA, USA (2002)
16. Xue, S.D., Zeng, J.C.: Swarm Robots Search under Conditions of Heterogeneous Sensory Signals Fusion. In: Proc. of the Int'l. Conf. on Genetic and Evolutionary Methods, Las Vegas, NV, USA, July 14–17, pp. 122–128 (2008)

An Incentive Mechanism for Increasing Repayment Rate for Online Social Lending

Masashi Iwakami[1], Takayuki Ito[2], and Joaquin Delgado[3]

[1] Nagoya Institute of Technology, Gokiso, Showaku, Nagoya 466-8555, Japan
iwakami@itolab.mta.nitech.ac.jp
[2] CCI, MIT Sloan School of Management, 5 Cambridge Center, Cambridge,
MA 02139, USA
takayuki@mit.edu
[3] Yahoo! Inc., 4401 Great America Pkway, Santa Clara, CA 95054, USA
jdelgado@yahoo-inc.com

Abstract. Social lending is, in its broadest sense, a certain breed of financial transaction (primarily lending and borrowing) that occurs directly between individuals without the intermediation or participation of a traditional financial institution. Social lending is a dynamic trading mechanism that can directly match one consumer with another consumer. Most manual transaction processes conducted by traditional financial institutions can be done automatically and tailored to each consumer in social lending so that every player can maximize his own values in the transactions. The processes that could be automated include adjusting interest rates, creating incentives for borrowers to return money, recommending lenders and borrowers, creating portfolios, etc. In this paper we focus on an incentive mechanism for borrowers because they are crucial for dynamic social lending since they could help increase worth and reduce payment delays. We propose an incentive mechanism that improves a borrower's payment delay score. The mechanism offers incentives for payment with rewards (penalties) to borrowers. We demonstrate the efficiency on our proposed methods by conducting agent simulations.

Keywords: Social Lending, Incentive Mechanisms.

1 Introduction

Social lending is, in its broadest sense, a certain breed of financial transaction (primarily lending and borrowing) that occurs directly between individuals without the intermediation or participation of a traditional financial institution. Online platforms such as eBay have successfully attracted individuals to interact on the Internet. This concept has spread to consumer lending. For example, the following companies are famous for social lending: Lending Club [1], PROSPER [2], Zopa [3][4], etc.

Social lending has been growing for the following reasons: (1) For lenders, providing money through social lending gives a higher interest rate than depositing with a deposit company. (2) For borrowers, borrowing money through social

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 566–575, 2009.

lending gives a lower interest rate than borrowing money from a loan company. (3) For borrowers, passing the loan application is easy. (4) There is a sense of security with investment choices based on the information disclosure of SNS(Social Networking Service). For reasons (1) and (2), such unnecessary expenses as brokerage fees are not required. For reason (3), there are many lenders. A bank is only one lender, but in social lending there are many lenders. The reason for (4) is that the past history and preferences of users can be seen in the SNS community. With additional information about borrowers, lenders can easily judge them.

In social lending, most manual transaction processes done by traditional financial institutions can be automatically conducted and tailored to consumers so that all players can maximize their own values. The processes that could be automated include adjusting interest rates, making incentives to borrowers for returning money, recommending lenders and borrowers, creating portfolios, etc. In this paper we focus on an incentive mechanism for borrowers because they are crucial for dynamic social lending and might help increase the value of players and reduce payment delays.

The remainder of this paper is organized as follows. First, we explain the related literature on offline group lending. Second, we focus on removing the return delay for borrowers after financing and propose a mechanism to groups for improving the return delay rate. Next, we present an experimental assessment of these models. Finally, we conclude with a discussion of possible avenues for future work.

2 Related Literature

2.1 Related Literature on Offline Group Lending

The related literature on offline group lending is plentiful. Based on the success of the Grameen bank [5], there have been many studies on group lending and group capital in development economic literature. It is well-known that physical group lending mechanisms in rural areas have shown high repayment rates. The Grameen bank is one famous example of such a mechanism.

Basically most papers explain why group lending can be successful. For example, [6] explained how joint-liability credit contracts used by group lending schemes can achieve high repayment rates even if borrowers have no collateral. [7] analyzed how joint-liability lending promotes screening, monitoring, state verification, and enforcement of repayment and shows two case studies in practice. [8] shows the improved results if the models incorporate the threat of not being refinanced if the group defaults; the result (positive assortative matching), obtained by [7], is not necessary. [9] argues about the importance of sequential financing utilized in the Grameen bank.

Table 1. Statistical information about delay (Sep 13, 2008)

Delay	People	Rate (%)	Accumulation ratio (%)
0	3457	84.77	84.77
1	414	10.15	94.92
2	132	3.24	98.16
3	44	1.08	99.24
4	17	0.42	99.66
5	5	0.12	99.78
6	2	0.05	99.83
7	3	0.07	99.90
8	2	0.05	99.95
9	0	0.00	99.95
10	0	0.00	99.95
11	1	0.02	99.98
12	0	0.00	99.98
13	1	0.02	100.00

3 Incentive Mechanism for Improving Delay Score

3.1 Improving Delay Score

This section introduces a mechanism as an incentive to decrease the delay frequency and to increase the borrower incentive to repay.

Table 1 is the statistical information disclosed by the Lending Club on its Web site. The content is the repayment delay frequency of a borrower for two years. The rate of borrowers with at least one late payment is about 15%. Even one such delay helps lenders imagine the consequences of payment failure, and they might feel that the investment is not attractive. If we can reduce one-time or two-time delays, the borrower-not-delayed ratio is about 98%, and the system will be more attractive for investments. We didn't consider one- or two-time delays critical because they might simply be borrower carelessness or negligence. If we can give a certain incentive to borrowers, such small delays might be distinguished. Therefore, an incentive mechanism is introduced to decrease the delay frequency and to encourage borrowers to repay.

When assuming a simple loan situation in which many lenders fund many borrowers without a reputation mechanism, a selfish borrower has an incentive to neglect and increase the number of delays. The reason is that for a lender who lends small amounts money to many borrowers, blaming one borrower out of all of the borrowers is not profitable. Therefore, no repayment effort is nearly a dominant strategy. In this case, only borrowers profit, not lenders. Thus, an incentive mechanism can be effective for evading such situations. We concentrated on the fact that borrowers can form groups mutually or automatically, which is a peculiar character with SNS. Using the idea of grouping, we designed an incentive mechanism that resembles a reward (penalty) mechanism. A similar type of grouping can be seen in offline lending mechanisms.

3.2 Designing an Incentive Mechanism

In this section, we design a mechanism that encourages borrowers to repay. It creates several groups of borrowers and gives a reward (penalty) to them based on payment results. The distance from the average delay rate of the group decides the amount of reward (penalty). For instance, if the delay rate of individual i is under the average, a reward is given to i. Oppositely, if the delay rate of individual i is over the average, a penalty is given to i.

Consider an economy of n individuals. Let $k = 1, ..., m$ be the groups in the economy, and let I_k be the set of individuals in group k. The mechanism owner makes individual i belong to group k. When the delay rate of i is g_i, reward (penalty) r_i of i is as follows:

$$r_i = P_G(g_i - \frac{1}{n_k - 1} G^k_{-i}) \tag{1}$$

where $G^k_{-i} \equiv \sum_{j \in I_k - \{i\}}^{n_k} g_j$, $n_k \leq n$ is the number of people belonging to group k, and G^k_{-i} is the sum of the total delay rate of the members of this group except the contribution of individual i. This general reward function is known as the Falkinger mechanism [10]. We instantiate P_G, which decides the reward and penalty value of the borrower.

The following are the characteristics of this mechanism. In this mechanism as well as the payment calculation of the VCG mechanism [11], [12], [13], the reward (penalty) is dependent on the results of others: G^k_{-i}. g_i cannot influence its own reward (penalty) in handling its repayment result. Thus, since borrowers don't have an incentive to intentionally deteriorate the repayment results, they will repay at the maximum effort. Also, the mechanism owner doesn't need a large budget, despite the value of P_G, which does not influence the financial condition of the mechanism owner. This is relatively obvious. Here $\sum_{i=1}^{n} r_i = 0$, since by definition for each group, $\sum_{i \in I_k} G^k_{-i} = (n_k - 1) \cdot \sum_{i \in I_k} g_i$.

The total amount of rewards and penalties is 0. They disappear if all borrowers achieve the same repayment results. As Table 1 shows, most borrowers have no delays. But the bottom 15% of borrowers have some delays. When borrowers who delay strive to avoid that situation, no one needs to pay a penalty under this mechanism.

3.3 Calculating Expected Loss for Owners

Our mechanism has one problem. The amount of penalty can't be increased, even if the entire repayment score falls because the mechanism decides the penalty by the distance from the average late payment of the group. For example, if the delay rates of all borrowers fall, there is no change in the total of the penalty. However, the earnings of the mechanism owner decreased due to worsening repayment scores, which will decrease the attraction of the site.

Thus we consider the owner's expected loss. The owner expects a (one-step) future loss each time and gets it as a tax from borrowers. When collecting the expected losses from groups, we include one agent without any delays in each group. For example, in a group composed of borrowers A and B, we set the agent who collects the group's expected losses. It is impossible for the agent to pay a penalty because it has no delay. When A's delay rate is 0.1 and B's delay rate is 0.2, the reward the agent gets is: $tax = P_G(0 - \frac{0.1+0.2}{2}) = P_G(-0.15)$. In this way, the agents are set into groups to collect taxes.

The expected loss is defined as E, which is derived from the delay results of all borrowers. v_k is defined as a tax rate collected by the agent from group k. v_k is not money but a rate. Taxes are defined as V_k. $V_k = \frac{v_k}{\sum_{i=1}^{m} v_i} E$. Let x be the reward rate. The following is the reward (penalty) money that the borrower or one loss collection agent receives:

$$P_G(x) = \frac{E}{\sum_{k=1}^{m} v_k} x. \tag{2}$$

The total sum collected by tax agents is $\sum_{l=1}^{m} P_G(v_l) = \sum_{l=1}^{m} \frac{E}{\sum_{k=1}^{m} v_k} v_l = \frac{E}{\sum_{k=1}^{m} v_k} \sum_{l=1}^{m} v_l = E$.

Let us define a simple function to model the expected loss. Suppose that lenders regard whether borrowers delay as important. In this case, if 100% of borrowers have no delay, lenders are attracted. With a decreasing rate of no-delay borrowers, lenders begin to avoid lending. For lenders, whether the ratio is 99 or 100% is crucial, but not whether the ratio is 0 or 1%. The width of the harm decrease when the ratio decreases from 100 to 99% is larger than that when the ratio decreases from 50 to 49%. Judging from the preferences of the lenders, their preference function increases monotonically, and the convex function is in the given domain. When x becomes the ratio of the 0-delay borrowers, investment participation rate R can be defined as follows:

$$R(x) = x^2 \quad (0 \leq x \leq 1). \tag{3}$$

Moreover, when x_t becomes the ratio of the 0-delay borrowers at period t, M is the average commission the owner gets from one lender, and N is the number of lenders in the system, the following are expected losses E:

$$E = -M \times N \times (R(x_t) - R(x_{t-1})). \tag{4}$$

Below is a simple calculation example. Imagine a case where the commission is 1%, one lender lent \$10,000, and the system has 1000 lenders. At the first term, the ratio of the borrower who has no delays is 95%, and in the second term, the ratio of the borrower who has no delays is 90%. The amount of expectation loss for the mechanism owner is $-(10000 \times 0.01) \times 1000 \times (R(0.9) - R(0.95)) = 9250\$$. Suppose the system has 100 borrower groups and that the reward rate received by the loss collection agent is 0.05 per group. The total tax rate collected by the agents is $0.05 \times 100 = 1.5$. The penalty money per 1% penalty rate is $9250 \div 1.5 \div 100 = 61.66\$$.

3.4 Grouping Effect

Borrowers with a high delay probability sometimes receive excessive penalties by the group condition even if they make a maximum effort under the mechanism. This happens when the payment abilities of borrowers are very different from each other in groups. Unlucky borrowers with slight payment ability have to pay more money if they belong to a group with a good score. In this section, we show an example of an undesirable grouping and a solution.

Assume a four-person economy. The borrowers are A, B, C, and D. All people maintain maximum effort. A and B have no delays. C and D have four delays with maximum effort. When a group is formed with 2-human units, there are two types of groupings:

1. {A,B},{C,D}
2. {A,C},{B,D}({A,D},{B,C}).

In the first grouping, no one pays money because the average delay of group {A,B} is zero and the average delay of group {C,D} is four. In the second grouping, the average delay of group {A,C} is two, and it is also two for group {B,D}. Therefore, C and D have to pay a penalty. It is not preferable to generate the penalty although four people make maximum efforts. The first grouping is better. Generally, avoiding the moving of money from person to person is better, assuming that everyone has made maximum effort. The best way to meet this requirement is to combine borrowers with mutually similar delay scores. The algorithm is as follows:

1. Divide borrowers into those who have payment histories and those who don't.
2. Sort the borrowers with histories by history scores and combine them into groups.
3. Randomly combine borrowers without histories.

By the above method, borrowers do not need to pay unsuitable penalties in groups where everyone has a history because in each group the payment scores are close to each other.

4 Experiments

4.1 Experiment on Incentive Mechanism

We experimentally evaluated our proposal's effectiveness by imposing the reward (penalty) mechanism. As settings, borrower agents were generated who have delay probability and internal effort level. The effort level affects the borrowers' delay probability. If the effort level is maximum, the agent decreases the delay probability by 0.1%. They can decide the effort level based on the mechanism. If there is no mechanism, the effort level is zero. Therefore, the borrower's delay probability has no change. If there is our mechanism, the effort level is maximum. The borrowers must repay ten times after their effort levels are determined. After

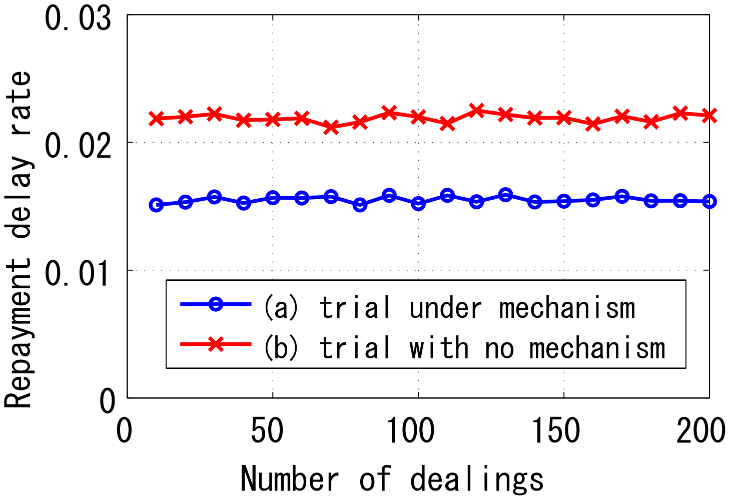


Fig. 1. Comparison of total payment delay rate

repayment ends, the system calculates the repayment delay rate of all borrowers and penalizes them based on the repayment results in the group to which each borrower belongs. In the actual experiment, the above operation was repeated 100 times. The average of 100 operations was adopted as the result. The following are the experiment settings.

- number of borrower agents: 1000
- ratio of agent (delay probability is 0.001): 85%
- ratio of agent (delay probability is 0.01): 10%
- ratio of agent (delay probability is 0.02): 4%
- ratio of agent (delay probability is 0.03): 1%

The following repayment delay rate is expected: $0.001 \times 0.85 + 0.1 \times 0.1 + 0.2 \times 0.04 + 0.3 \times 0.01 = 0.02185$. When all borrowers repay, the expected repayment delay rate will be $0.01 \times 0.85 + 0.01 \times 0.1 + 0.1 \times 0.04 + 0.2 \times 0.01 = 0.0155$, due to 0.1% repayment decreases in the borrower delay rate.

Fig.1 shows the average of the repayment delay rate. The vertical axis is the average of the repayment delay rate. The horizontal axis is the dealings frequency. (a) shows the payment delay rate under our mechanism, and (b) shows the payment delay rate with no mechanism. The delay rate of graph (a) is lower than that of graph (b) because the borrowers who belong to the group had to make repayment efforts and improve their delay rate. As expected, the result value of (a) is lowered near the theoretical value. These results show that we decreased the repayment delay rate of the borrowers.

4.2 Comparison of Payment Penalty Amount

In this section, we confirm the effectiveness of the grouping of the borrowers by comparing two situations: when the group is composed randomly and when the

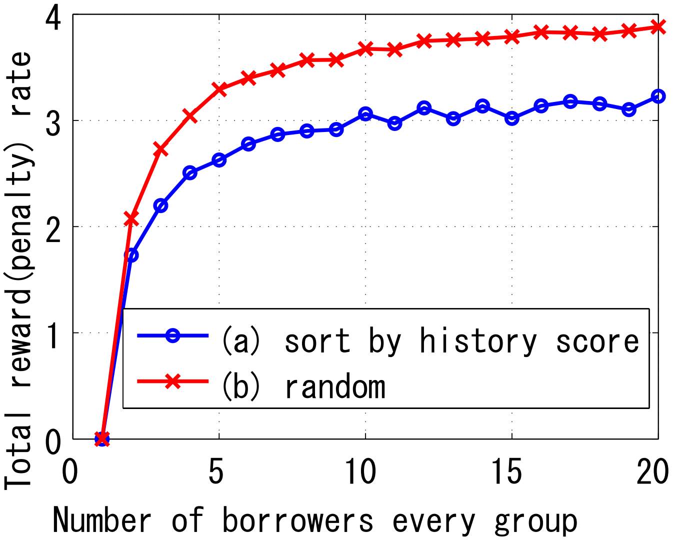


Fig. 2. Comparison of total payment penalty

system chooses the group intentionally. The system divided borrowers into two groups: with and without payment histories. Next, the system sorts the former in order of return results and then makes groups by that order. The latter is grouped randomly because it cannot be sorted since it has no payment history.

Fig.2 shows a comparison of the total payment penalty rate of the borrowers. The vertical axis is the total payment penalty rate of the borrowers. The horizontal axis is the number of borrowers in every group. (b) shows the total payment penalty when the group is composed randomly, and (a) shows the total payment penalty when the system intentionally composes the group. Let us compare the case when the groups are comprised of five people. For (b), the total reward (penalty) rate was around 3.3% because the repayment delay rate varies every group, and the penalty is paid. For (a), the total reward (penalty) rate was around 2.6% because the repayment ability is balanced in every group and borrowers pay little penalty. Whatever the number of borrowers in every group is, making groups by sorting lowers the penalty rate that doesn't need to be paid by borrowers.

4.3 Experiment of Profit Loss Compensation

In this section, we observe the transition of owner earnings by the compensation mechanism. The borrowers' composition is the same as in the previous section. Assume about 1000 lenders are in the system. The lender finances an average of $10,000, and the commission is 1%. The lender judges whether to invest based on the proportion of no-delay borrowers. The following is the investment participation rate of the lenders: $R(x) = x^2$, where x is the proportion of no-delay borrowers.

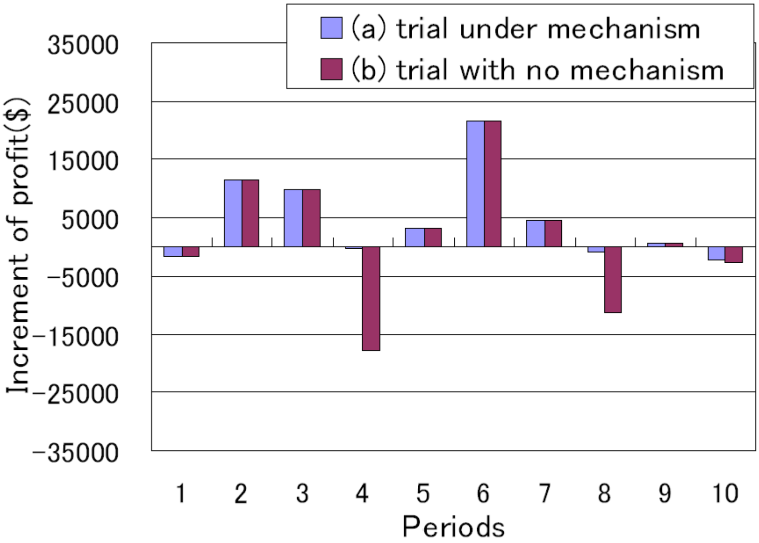


Fig. 3. Comparison of profit increment

The owner calculates the expected loss at period $t+1$ with the following expression: $E_{t+1} = -0.01 \times 1000 \times (R(x_t) - R(x_{t-1}))$. Under the compensation mechanism, owners can compensate for the earnings of the next period because they can predict the expected loss, which they can collect from the borrowers.

Fig.3 shows the transition of the profit increment. The vertical axis shows the profit increment, and the horizontal axis shows the number of periods. (b) shows profit increment without compensation, and (a) shows profit increment to which the compensation was applied. Graph (b) fluctuates and sometimes the profit decreases, but graph (a) is not less than zero very much. The reason is that the difference of earnings is suppressed by compensating for the money collected from the borrower using the compensation mechanism. This also shows that the expectation is functioning well.

5 Conclusion and Future Work

Social lending is forming a new financial transaction mechanism on SNS. Due to its dynamic nature, which has not previously existed in traditional financial transactions, many fields need to be clarified. In this paper, we proposed a mechanism that improves payment delay scores by creating an incentive for payment with rewards (penalties). Borrowers are encouraged to repay under our mechanism, and average delay scores improve. By sorting borrowers into groups, the penalties are lowered that don't need to be paid by borrowers. In addition, by setting the expected loss collection agents into groups, the mechanism owner can avoid profit loss when the entire delay score of the borrowers decreases.

Our future work includes comparing our incentive mechanism with another incentive mechanism and scaling up the experiments.

References

1. Lending Club, http://www.lendingclub.com/
2. Prosper, http://www.prosper.com/
3. Zopa UK, http://uk.zopa.com/ZopaWeb/
4. Zopa USA, https://us.zopa.com/
5. Grameen Bank, http://www.grameen-info.org/
6. Ghatak, M.: Group lending, local information and peer selection. Journal of Development Economics 60, 27–50 (1999)
7. Ghatak, M., Guinnane, T.W.: The economics of lending with joint liability: Theory and practice. Journal of Development Economics 60, 195–228 (1999)
8. Guttman, J.M.: Assortative matching, adverse selection, and group lending. Journal of Development Economics 87, 51–56 (2008)
9. Chowdhury, P.R.: Group-lending: Sequential financing, lender monitoring, and joint liability. Journal of Development Economics 77, 415–439 (2005)
10. Falkinger, J., Fehr, E., Gachter, S., Winter-Ebmer, R.: A simple mechanism for the efficient provision of public goods: Experimental evidence. The American Economic Review 90, 247–264 (2000)
11. Vickrey, W.: Counterspeculation, auctions, and competitive sealed tenders. The Journal of Finance 16, 8–37 (1961)
12. Clarke, E.H.: Multipart pricing of public goods. Public Choice 18, 19–33 (1971)
13. Groves, T.: Incentives in teams. Econometrica 41, 617–631 (1973)

A Platform System for Developing a Collaborative Mutually Adaptive Agent

Yong Xu[1], Yoshimasa Ohmoto[1], Kazuhiro Ueda[2], Takanori Komatsu[3],
Takeshi Okadome[4], Koji Kamei[4], Shogo Okada[1],
Yasuyuki Sumi[1], and Toyoaki Nishida[1]

[1] Graduate School of Informatics, Kyoto University,
Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan
[2] Department of System Sciences, the University of Tokyo
[3] International Young Researcher Empowement Center, Shinshu University
[4] Innovative Communication Laboratory,
NTT Communication Science Laboratories
xuyong@ii.ist.i.kyoto-u.ac.jp,
{ohmoto,okada_s,sumi,nishida}@i.kyoto-u.ac.jp,
ueda@gregorio.c.u-tokyo.ac.jp, tkomat@shinshu-u.ac.jp,
houmi@idea.brl.ntt.co.jp, kamei@cslab.kecl.ntt.co.jp

Abstract. The characteristic task of service robots that can interact with humans is to achieve human-robot collaboration. Mutual adaptation is considered to be an important characteristic of robots, required for carrying out such collaborative tasks. Here, we introduce the concept of mutual adaptation, propose a learning model, and describe an experimental task to explain the above concept. A waiter robot performs a collaborative task using a platform system, which is developed by a constructive approach. The interactive and manual modes of this system are compared by performing a preliminary experiment to evaluate the effectiveness of the robot's autonomous function. The results indicate that the robot's autonomous function works well when operated in the interactive mode under short time or slow speed conditions.

Keywords: Human Robot, Intelligent Interfaces, Mutual Adaptation.

1 Introduction

The characteristic task of service robots that can interact with human users is to achieve human-robot collaboration. At present, research is being conducted on developing an intelligent robot (agent) that is capable of receiving and learning undefined instructions from humans. Thomaz et al. [1] developed an interactive computer game system known as Sophie's Kitchen, in order to enable humans to teach a robot and simultaneously improve the robot's learning behavior. From the results of their research, it was found that humans dynamically change their behavior as they develop a mental model of a robotic learner. However, their study mainly focused on a reinforcement learning task where humans know the

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 576–585, 2009.

goal of a task which is not known by the robot; in such a situation, humans often take initiative. Yamada et al. [2] proposed the concept of "adaptation gap." They argued that there often exists an adaptation gap between the expected ability and real ability when human users and robots try to build an internal prediction model for each other. The results of their study suggest that in the case of collaborative agents, especially agents that are required to interact with humans, it is very difficult to develop a completely autonomous agent that can accomplish collaborative tasks by adapting to humans and simultaneously ensure that humans can easily teach the robotic learner. Authors argue that mutual adaptation is an important characteristic of collaborative robots. Service robots apparently differ from traditional industry robots because they are generally required to interact with human users in an uncontrolled environment. Among various types of tasks of service robots, the task of a waiter robot is a typical task.

2 Concept of Mutual Adaptation

2.1 Definition and Hierarchical Structure

First, it is assumed that there are two agents A and B with different abilities; A and B perform the same collaborative task. In this task, A and B can only share partial information with each other. In order to achieve a common purpose, each agent is required to build a model for the other, develop a communication protocol that can be used to interpret information from the other, and draw inferences about the other depending on various situations. In general, the agents have to complete the task by gradually adapting to each other. In this paper, the above-mentioned phenomenon is termed *mutual adaptation*.

Mutual adaptation is observed in human-human communication. Xu et al. [3] performed an experiment, the result of which indicates that mutual adaptation has a hierarchical structure. Xu et al. [4] developed a two-layered model, which indicates that mutually adaptive behaviors can take advantage of at least two levels of mutual adaptation protocols. In the lower level, basic protocols are developed using fundamental elements. In the upper level, abstract protocols are developed by combining elements from the lower level or creating new elements.

2.2 Preconditions and Occurrence Conditions

In order to design an experiment to invoke mutual adaptation, it is necessary to define the preconditions of mutual adaptation, which are summarized as follows.

- Asymmetric ability: The abilities of A and B are said to be asymmetric if the instructor agent is a human and the learner agent is a robot. Both the agents require the following abilities to perform the task in this study:

$$A_{human} = A_{common} + \alpha \tag{1}$$

$$A_{robot} = A_{common}, \tag{2}$$

where A_{human} represents the ability of the human, A_{robot} represents the ability of the robot, and A_{common} represents the abilities common to both the human and the robot.

(1)Abilities common to both the human and robot

- Ability to understand rewards received from the partner
- Ability to take optimal actions
- Ability to take initiative (autonomous actions)

(2)Ability of α of the human instructor

- Ability to develop new concepts
- Ability to propose new protocols
- Ability to build a model for the entire human-robot collaborative system

Although the robot developed by present technologies may lack the ability to mutually adapt, it is still necessary to clarify how humans with the above-mentioned abilities are capable of mutually adapting.

- Asymmetric information: When A and B obtain only asymmetric information, it implies that each agent does not have complete information that is necessary for completing the collaborative task.
- Sharable entire evaluation score: It is necessary to provide an entire evaluation score that both A and B can obtain simultaneously. In order to get a high score, it is necessary to design a suitable task, which cannot be accomplished without successfully sharing information, and developing and exchanging protocols (communication) between the two agents.
- Transition initiative: In some situations where A acquires sufficient information, A may always take initiative and instruct B to follow its instructions. However, in some other situations where B has some information that A does not have, or B knows that by performing an autonomous action, a high score can be achieve, B may perform the task on its own, so that the initiative is transferred from A to B.

In general, the sufficient and necessary conditions can be expressed as $X \rightarrow Y$, which implies that X is the sufficient condition of Y, and Y is the necessary condition of X.

Sufficient condition: If the following conditions are satisfied, there is a high possibility for the occurrence of mutual adaptation.

- The human instructor is aware of the merit of robot's autonomous action.
- The human instructor changes his/her strategy or instruction approach to adapt to the robot.

Necessary condition: If a mutual adaptation phenomenon occurs, there is a high possibility for the occurrence of the following:

- Each agent understands the behaviors (instructions or actions) of the other, irrespective of whether the partner was aware of it or not.
- The information shared is asymmetric.
- Both the agents take initiative.
- Common protocols are developed.

On the basis of the above-mentioned definition and occurrence conditions of mutual adaptation, it is necessary to verify them by performing a specific human-robot collaborative task.

3 Experiment to Clarify Mutual Adaptation

3.1 Objective

The objective of this study is to explain the concept of mutual adaptation by considering a training task of a waiter robot; in this task, a manager (human instructor) trains a waiter (robot) in a restaurant. It is essential that the degree of satisfaction of the customers and the profit of the restaurant are maximized simultaneously. Here, the degree of satisfaction of the customers is reflected by the amount of tip given to the waiter robot.

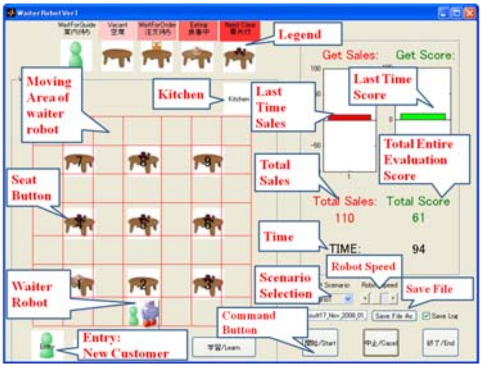
The objective of this study is to enable instruction-based learning, but not model-based learning. This is because the meaning of the same instruction can be interpreted differently, depending on different situations. This study focuses on not only learning new environments, but also learning new possible instructions. To obtain an ideal result, human users are expected to be aware of the merit of autonomous actions performed by the robot; on the basis of the obtained result, humans give the same instructions implying different meanings under different situations. On the other hand, robots are expected to respond to instructions properly by developing common protocols, adapting to changes, and demonstrating their capabilities in learning human models, so that mutual adaptation can be invoked.

3.2 Task

The waiter robot performs its task by using a software simulation system. The task is to be performed by a manager (human instructor), waiter (software robot), and some customers (generated and controlled by an automatic program). The table layout is shown in Fig. 1. Each table has up to the following five states: “vacant table,” “waiting for guide (only available for entrance),” “waiting for order,” “eating,” and “clearing table.”

3.3 Model

There are two types of instructions that can be given by the manager, namely, action instruction and evaluation instruction. The manager can decide the target table for the robot by referring to the current position of the robot; he can provide the seat number of the destination to the robot in the form of instructions. The objective of the task is to maximize the sales of the restaurant (i.e., minimize the total stay time of all the customers) and simultaneously maximize the scores of the entire evaluation function (i.e., maximize tips received by the robot and the sales of the restaurant) by minimizing the movement time of the robot. On



Total Sales: 110
Total Score 61
TIME: 94

Fig. 1. Graphical User Interface of the waiter robot simulation system

the other hand, the robot simultaneously aims to maximize the tips received from the customers and the sales of the restaurant by minimizing the customers' waiting time before they are guided to a seat or before the robot takes their order.

In principle, the waiter robot has two types of choices i.e., it can move by following the manager's instruction, or it can move autonomously by complying with its build-in rules or by using previous knowledge. For example, the robot may move towards a customer who is sitting at the table nearest to its current position and waiting to place his order, or it may move along the shortest route to provide services to two or more customers who are waiting to be served. In general, the robot always tries to move along the best route, in order to receive maximum tips and reduce the waiting time of the customers'. The robot needs to strike a balance between maximizing tips and minimizing waiting time. In order to achieve the former, the robot moves along a complex route to provide services to the maximum possible customers. However, in order to achieve the latter, the robot moves along a simple route to a customer who has been waiting to be served a long time. A customer who has been waiting for a very long time may leave the restaurant, causing a loss of sales and revenue; therefore, it is necessary for the robot to clear the tables immediately after the customers leave, so that there is a sufficient number of available vacant tables for new customers.

There are three types of rewards, namely, sales, tips, and scores. The amount of restaurant's sales is a type of reward that the manager aims to maximize. This amount relates to the number of customers who place orders and do not leave. The amount of tips received by the robot can only be known by it. The robot always tries to maximize the tips it received. Changing trends in scores are considered to be a common reward that is shared by both the manager and the waiter robot.

The relationship between the three evaluation functions is shown in Fig. 2. The first function of tips is only applicable to the robot; this function enables the robot to choose its default action depending on the expected values of the tip. On the other hand, the human instructor, who plays the role of a manager,

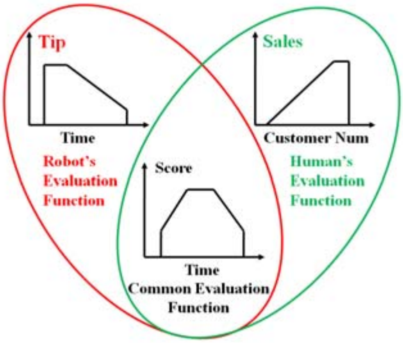

Tip
Time
Robot's Evaluation Function
Sales
Customer Num
Human's Evaluation Function
Score
Time
Common Evaluation Function

Fig. 2. Evaluation Function

has no knowledge of the amount of tip received; therefore the manager prefers to instruct the robot to serve the table nearest to its current position. The entire evaluation function is assumed to be shared by the robot and the manager.

The learning function will be implemented by using the model as shown in Fig. 3. This model mainly consists of the following four modules: human instruction evaluator (HIE), robot action evaluator (RAE) , robot action generator (RAG) , and robot action decision-maker (RAD). The main functions of the HIE and RAE modules are to evaluate the manager's instruction and robot's action; the score changes on the basis of the result of the evaluation. The RAD module compares the results obtained by the above two modules and decides the type of action that the robot is required to perform; the actions are defined in the RAE and RAG modules. The results of the real robot action are provided to both the RAE and RAG modules, so that they can be used as references for learning in future. The score calculation function is used to generate the score for the simulation system; however, this function is not known by the robot. If the robot has knowledge about the score, it can always take optimal action without receiving any instructions from the manager.

In order to develop such an experimental system, which is expected to be able to invoke mutual adaptation, it is necessary to develop a platform system where the autonomous function of the robot can work well to generate sufficient scores without receiving any instructions from the manager. In order to evaluate the effectiveness of the robot's autonomous function, a preliminary experiment is performed. The details of this experiment will be described in the next section.

3.4 Method

A waiter robot simulator software system with a graphical user interface was developed, as shown in Fig. 1. Human participants can instruct the robot by pressing buttons. They are informed in advance that the robot can perform some autonomous actions based on its own decisions. The goal of the task is to maximize the sales and the scores within a limited time. First, the platform

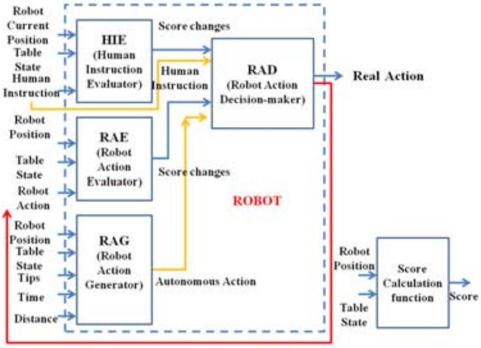


Fig. 3. Learning Model

system with an autonomous robot capable of interactive responses needs to be tested. If the autonomous function is very poor, it may become necessary for the manager to intervene regularly. On the other hand, if the function is very perfect, the manager may issue instructions and then simply observe the robot performing its tasks without giving any further instructions. Both these cases are unsuitable for invoking mutual adaptation.

A typical scenario is as follows. When a new customer enters the restaurant, if a seat is vacant and the robot is idle, it should move to the entrance and guide the customer to the vacant seat. The waiter robot is assumed to have three states:"normal," "guiding customer," and "carrying dish." When it is guiding a customer, its state will change to "guiding customer" automatically, and until the customer is seated, the manager's instructions will be ignored. If there is neither an available vacant seat nor a seated customer waiting to place an order, the robot needs to find a "need to clear" table, move there, clear the table, and change its state to "carrying dish" while carrying dishes to the kitchen. However, if any customer is waiting to place an order, the robot should first take the order by obeying its tip-first law. After the customer has been seated at a vacant seat, the table state will be changed to "waiting for order" state. It will be kept unavailable for placing orders for a specific period. In order to prevent customers from leaving the restaurant because they have waited too long time, which in turn reduces the sales and scores, the robot always tries to place an order as soon as possible when operating in the interactive mode. When the robot finishes placing an order, it receives tips, and the restaurant's sales and scores are updated simultaneously. All these data are recorded into log files.

In order to verify the effectiveness of the robot's autonomous function, an interactive mode is implemented in addition to the manual mode. Since the goal of this system is to invoke mutual adaptation, it is reasonable to expect the robot to obtain a sufficiently high score through its autonomous function. It is expected that the manager's method of instructing will be affected by the robot's autonomous function. This in turn will enable the robot to learn and improve its performance and adapt to the manager by trying various actions. Therefore, it is desirable to invoke mutual adaptation.

Table 1. Comparison Results for Manual & Interactive Mode with Statistical Value

TrialNo	RunningMode	Speed	Time(s)	N	GetScoreMean(S.D.)	GetSalesMean(S.D.)
1	manual	fast	60	11	76.69(14.10)	159.72(29.44)
1	interactive	fast	60	11	84.85(16.88)	178.09(35.22)
2	manual	fast	60	9	73.87(6.93)	139.67(17.31)
2	interactive	fast	60	9	84.49(3.69)	166.22(9.23)
3	manual	fast	300	9	273.80(37.50)	562.22(80.69)
3	interactive	fast	300	9	257.76(23.96)	535.44(50.37)
1	manual	slow	60	9	39.9(18.36)	82.33(38.58)
1	interactive	slow	60	9	56.13(13.49)	117.0(27.97)
2	manual	slow	60	10	38.08(13.12)	82.7(22.17)
2	interactive	slow	60	10	56.46(9.56)	123.4(15.06)
3	manual	slow	300	1	173.6	344.0
3	interactive	slow	300	1	183.0	375.5

3.5 Results and Discussions

A software platform for the waiter robot simulator system was developed, and among the 10 scenarios available, 3 were tested. Two modes of the robot, namely, the manual and interactive modes, were implemented in this system. In the manual mode, the waiter robot cannot move without receiving instructions from the manager. In contrast, in the interactive mode, the robot can move autonomously without requiring instructions from the manager or having to change its own actions to follow the manager's instructions. The robot can move at fast (one step per 0.5 s) or slow speeds (one step per 1 s). Two types of user interfaces, namely, a touch panel and a mouse, can be used to control the robot.

10 students (9 male and 1 female, age: 22-33 years, average age: 26) participated in this experiment.First, the participants were asked to complete the first trial using both the touch panel and the mouse at fast and slow speeds. Then, they were asked to complete the second trial using their preferred user interface at fast and slow speeds. Finally, they were asked to complete the last trial using their preferred user interface and speed. In each trial, the participants were asked to finish the task in both the manual and interactive modes. The results of all trials including the scores, sales, robot states, table states, and the manager's press-button actions were recorded in log files.

The results of the mean values of the scores and sales achieved by the participants during the experiment are listed in Table 1.

A series of two-sided paired-sample T-tests are applied to the experimental results to elucidate the differences between the manual and interactive modes. We consider the following null hypotheses for the three trials: there is no significant difference between the score and sales achieved by instructors when performing the task in the manual and interactive modes. The results of the T-test are listed in Table 2.

Table 2. Comparison Results for Manual & Interactive Mode Using Paired T-test

TrialNo	Speed	Time(s)	GetScore	GetSales
1	fast	60	m-mode<i-mode,t(10)=1.0 n.s.	m-mode<i-mode,t(10)=1.1 n.s.
2	fast	60	m-mode<i-mode,t(8)=3.6 **	m-mode<i-mode,t(8)=3.6 **
3	fast	300	m-mode>i-mode,t(8)=1.79 n.s.	m-mode>i-mode,t(8)=1.4 n.s.
1	slow	60	m-mode<i-mode,t(8)=4.396 **	m-mode<i-mode,t(8)=4.501 **
2	slow	60	m-mode<i-mode,t(9)=3.71 **	m-mode<i-mode,t(9)=4.63 **

(**:p< 0.01,*:p<0.05,+:p< 0.1,n.s.:p> 0.1,
m-mode:manaul mode,i-mode:interactive mode,df:degree of freedom)

With regard to the results, in the case of trial 1, the "score" and "sales" values differed significantly only under the slow speed condition, but not under the fast speed condition. In the case of trial 2, the two values differed significantly under both speed conditions. However, in the case of trial 3, no significant difference between the two values was observed.

The statistical analysis results indicate that the performance of the robot's autonomous function is better under short time trials or slow speed conditions than under long time trials or fast speed conditions. This implies that the robot's autonomous function works well in at least some situations.

On the basis of only the current results, it is difficult to confirm whether mutual adaptation is affected by the robot's autonomous function. Without considering the structure of the task, it may be difficult to understand why the performance of the interactive mode differed significantly from that of the manual mode in short time or slow speed trials, but not in long time or fast speed trials. We believe that in a collaborative task, it will be necessary for the human instructor to integrate his/her instruction model with the robot's autonomous functions to improve performance. In the interactive mode, some instructors issued few instructions and spent more time observing the robot's autonomous actions. This may imply that they tried to understand the intentions of the robot. In contrast, some instructors issued many instructions to the robot to make it follow their intentions. The current version of the system only allows for very simple autonomous functions, and therefore, it might not work effectively in long time trials. In order to improve the performance, it would be necessary for the robot to integrate the human instructor's instruction model with its own autonomous ability using some learning method. We believe that our mutual adaptation model is a solution to this problem. 8 out of 10 participants used both user interfaces (touch panel and mouse) in the trials; of these, only 3 preferred the touch panel. With regard to the robot's movement speed, 90% of the participants preferred the fast speed. Therefore, these parameters will be set as default parameters in future experiments.

A learning model is being developed based on the autonomous function of the robot in the platform system to enable it to learn from collaborative experience. Further, the participants' instruction behaviors will be analyzed to investigate how human instructors change their instruction methods. On the basis of these results, a typical nonparametric technique known as the PNN (*Probabilistic Neural Network*) model introduced in [5] will be used to implement a learning function for the robot. Since this system is implemented in the form of a game, Game theory [6] is expected to provide some theoretical references that will help formalize the problem. By combining game theory and machine learning, it may be possible to invoke multistage mutual adaptation.

4 Conclusion

This paper introduces the concept of mutual adaptation, proposes a learning model, and describes a task to explain the concept of mutual adaptation. A platform system of a collaborative waiter robot task is developed.In a preliminary experiment, two modes, namely, interactive and manual modes, of the system are compared to evaluate the effectiveness of the robot's autonomous function. The results indicate that the robot's autonomous function works better in interactive mode than in the manual mode under some conditions.

References

1. Thomaza, A.L., Breazeal, C.: Teachable robots: Understanding human teaching behavior to build more effective robot learners. Artificial Intelligence 172(6-7), 716–737 (2008)
2. Seiji, Y., Koh, K., Takanori, K.: Mutual adaptation and adaptation gap between a human and an agent. Journal of Japanese Society for Artificial Intelligence 21(6), 648–653 (2006)
3. Xu, Y., Ueda, K., Komatsu, T., Okadome, T., Hattori, T., Sumi, Y., Nishida, T.: Woz experiments for understanding mutual adaptation. Journal of AI& Society 23(2), 201–212 (2009)
4. Xu, Y., Ohmoto, Y., UedaA, K., Komatsu, T., Okadome, T., Kamei, K., Okada, S., Sumi, Y., Nishida, T.: Two-layered communicative protocol model in a cooperative directional guidance task. In: The 7th International Workshop on Social Intelligence Design, San Juan, Puerto Rico (2008)
5. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience Publication, Hoboken (2001)
6. Leyton-Brown, K., Shoham, Y.: Essentials of Game Theory - A Concise, Multidisciplinary Introduction. Morgan & Claypool, San Francisco (2008)

A Simple Method of Forecasting Option Prices Based on Neural Networks

Xun Liang, Haisheng Zhang, and Xiang Li

Institute of Computer Science and Technology, Peking University, Beijing 100871, China
{liangxun,zhanghaisheng,lixiang}@icst.pku.edu.cn

Abstract. Options are an important financial derivative for the investors to control their investment risks in the security markets. The forecasting activity should realistically identify the option price in the future without knowing underlying asset price in advance. In this paper, a simple method of forecasting option prices based on neural networks is presented. We modify the traditional option pricing methods, enabling them to be eligible for forecasting the option prices. Then we employ the neural networks to further decrease the forecasting errors of the modified conventional methods. Finally, the experimental studies are conducted on the data of the Hong Kong option market, and the results demonstrate that the neural networks are able to improve the forecasting performance considerably. Conclusively, our neural network methods on option forecasting are fairly effectual in practice.

Keywords: Option prices, forecasting, Hong Kong option market, neural networks.

1 Introduction

In recent years, options have gained a conspicuous popularity in the security markets and have attracted numerous investors and speculators to accommodate the options in their portfolios. However, the investors' portfolios might be severely subject to jeopardy if they are unable to manage their options in a scientific way. It is therefore an indispensable step for the investors to establish justifiable and effective strategies while trading options.

Among the option pricing methods, the Black-Scholes model is mostly influential. The model prices the options based on the hypothesis that the underlying asset price follows a lognormal distribution [1]. In the last several years, many scholars also turned to the non-parametric methods such as the neural network (NN) methods. Intuitively, the simplest approach to use the NN is to estimate the whole option pricing function directly from the data [2]. However, the experiments demonstrated that the standard NNs do not performed satisfactorily because of a combination of factors, such as nonstationarity and noise of data [2]. To achieve a better performance, some scholars combined the conventional pricing model and the NN, and introduced a bootstrap approach to predict the difference between conventional parametric models and observed option prices. Also, [2] improved his method and presented a hybrid approach to enhance the performance including option boundaries including the

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 586–593, 2009.

scenario of deep-in-the-money. In [3], as opposed to learning the complicated option market straightforwardly, to acquire a more accurate result in the option market, the NNs are implemented upon the results of the well-known successful traditional methods. Since the platform of the traditional methods has already largely managed the nonstationarity and noise of the data, as well as the skew case of the deep-in-the money options, the burdens of NN approaches are greatly alleviated. [3] finally implemented the experiments on the option pricing for the Hong Kong option market, and a prominent improvement on pricing precisions over any of the conventional methods was archived.

The reminder of the paper is organized as follows. Section 2 modifies the four conventional option pricing approaches, enabling them to be qualified for forecasting the option prices. Section 3 presents two forecasting approaches by constructing the NNs on the platform supported by the conventional methods. Section 4 covers the empirical studies using data from the Hong Kong market. Finally, section 5 concludes the paper.

2 Option Price Forecasting by Modifying Conventional Approaches

The successful conventional option pricing methods used in this paper include the binomial tree (BT) method, the Black-Scholes (BS) method, the finite difference (FD) method, and the Monte Carlo (MC) method. Since all these methods are for determining the present theoretical option prices given the present underlying asset prices, for the purpose of forecasting the option prices at the end of next period, some modifications should be made.

In the modified convention models, the following notations are used, the underlying asset price *S*, the option's exercise price *X*, the current risk-free interest rate *r*, the asset price volatility σ, the time interval (length of a time period) Δt, the expiration time *T*, the option price *C*, and the forecasted option price $\hat{C}$.

2.1 A Modified Binomial Tree (BT) Method to Forecast Option Prices

The BT method assumes that within Δt, the underlying asset price $S_{t,\,i}$ (t = 0, 1, …, n-1) either goes up to $S_{t+1,\,i}$ (t = 1, …, n) according to a certain rate $u = e^{\sqrt{\sigma^2 \Delta t + (r\Delta t)^2}}$, or goes down to $S_{t+1,\,i+1}$ (t = 1, …, n) according to another rate $d = e^{-\sqrt{\sigma^2 \Delta t + (r\Delta t)^2}}$, where σ can be obtained from the asset's historical prices, i = 0, 1, …, n.

The BT contains t periods and (n+1) possible asset prices $S_{n,\,0}$, $S_{n,\,1}$, …, $S_{n,\,n}$ at the expiration time. There are two scenarios when the option is expired at the end of the last period. In the first scenario, its price at that moment is the difference between the asset price $S_{n,\,i}$ and the exercise price *X* if the option is in-the-money. In addition, noticing that option is a right other than an obligation, for an example of call option, when the asset price is lower than the exercise price, the investor has the right to choose not to exercise the option, thus making the option price 0. In summary of the above two scenarios, the option price is, $C_{n,\,i} = \max\{\ 0,\ S_{n,\,i} - X\ \}$, i = 0, 1, …, n,

$C_{t,i} = \max\left\{S_{t,i} - X,\ e^{-r\Delta t}\left[qC_{t+1,\,i} + (1-q)C_{t+1,\,i+1}\right]\right\}$, where q is the risk-neutral probability.

2.2 A Modified Black-Scholes (BS) Method to Forecast Option Prices

The BS method [1] first assumes that the movement of the asset price exhibits the lognormal distribution, and then utilizes the Ito's Lemma to obtain the option price as $C = f_{\mathrm{BS}}(S, X, t, r, \sigma)$. When the BS method is employed to predict the option price at the end of the next period, it is required to first estimate the corresponding asset price S_1. Here we choose a simple approach to achieve the estimation by $S_1 = e^{r\Delta t}S$. Subsequently, the forecasted option price at the end of the next period based on the BS method can be obtained as $\hat{C} = f_{\mathrm{BS}}(S_1, X, t\text{-}\Delta t, r, \sigma)$.

2.3 A Modified Finite Difference (FD) Method to Forecast Option Prices

The FD method employs a numeric approach to compute the option price, which is carried out via solving some differential equations with respect to the option. The essence of this approach lies in the transformation from differential equations into difference equations, which thereafter are solved in an iterative fashion by computers.

The conventional FD method segments T into a few time periods with the identical size Δt, and also segments the asset price, whose minimum is 0 and maximum $S_{\max}$, with the interval as ΔS. The computing process can be illustrated by a rectangular grid. The horizontal line represents the number of periods, while the vertical one the prices. Each node contains the underlying asset price and the option price.

In order to apply the FD method into option forecasting, we first establish an option price grid based on the FD pricing method. Second, we estimate the asset price in the next period in a similar way as we do with the BS method, namely $S_1 = e^{r\Delta t}S$. Third, we find in the second column of grid for the nodes closest to S_1. We use the interpolation method and locate the first sampling point, whose coordinate is formulated as $i = \left\lfloor \frac{S_1}{\Delta S} \right\rfloor$, where ΔS involves the granularity for calibrating the option price, $\lfloor \bullet \rfloor$ denotes the floor function. Note that the coordinate of the second sampling point is i+1. The proportion of interpolation can be calculated as $\chi = \frac{S_1}{\Delta S} - i$. We can therefore forecast the option price in the second period based on the FD method as $\hat{C} = (1 - \chi)C_i + \chi C_{i+1}$.

2.4 A Modified Monte Carlo (MC) Method to Forecast Option Prices

The MC method assumes that the asset price exhibits a lognormal distribution, which enables that the asset prices at the expiration time can be found by "averaging" a large of number of stochastic sampling trial paths. By discounting the average result back to the present, the corresponding option price is obtained.

After we have acquired the present option price C using the standard MC approach, the forecast counterpart at the end of next period can be calculated by $\hat{C} = e^{r\Delta t}C$.

3 Improving Forecasting Precisions Based on Neural Networks

The linear neural network (LNN) is adopted to learn the forecasting errors resulted from the four modified approaches compared to the real option prices.

The input vector of the LNN consists of the forecasted prices produced by the four aforementioned methods, $\left(\hat{C}_1(t),\ \hat{C}_2(t),\ \hat{C}_3(t),\ \hat{C}_4(t)\right)^\tau$ where $\hat{C}_1(t)$, $\hat{C}_2(t)$, $\hat{C}_3(t)$, and $\hat{C}_4(t)$ (t = -L, …, 0) are the forecast option prices at the end of period t, and $(\bullet)^\tau$ representing the transpose of (•). The subscript 1 denotes the BT method, 2 the BS method, 3 the FD method, and 4 the MC method, respectively.

The target output of the LNN is $C(t)$, where $C(t)$ (t = -L, …, 0) is the real option price at the end of period t. The activation function in the output layer is linear, and the training algorithm is the back propagation. The relation can therefore be depicted as $\hat{C}(t) = \sum_{i=1}^{4} w_i \hat{C}_i(t) + b$. Noticing that the option prices are most likely influenced by their recent historical prices, a dynamic sliding window of length L is designed to accommodate the option's historical prices and forecasted prices. In training, t is sequentially assigned the values from -L to 0, namely, the training set is in a dynamic fashion.

After the network has been trained, the forecasted prices for the next period, $\hat{U}'$ are input into the network in order to implement a new forecast $\hat{V}$ = $\hat{C}(1)$ for the next period.

Too, the multilayer perceptron (MLP) can also be utilized in forecasting option prices based on the modified the four conventional forecast results. The training and the forecasting procedures in the MLP resemble those in the LNN, whereas the difference in between lies in that the MLP includes a hidden layer and nonlinear activation functions, which enables it to learn the nonlinear features which are not "absorbed" by the modified conventional methods. Since it is well-known that the option market is a complex nonlinear system, the option price, a typical financial index of the option market, is supposed to exhibit the nonlinear features as well. In theory, the MLP is therefore preferable to the LNN.

4 Experiments

In this section, the time period is assumed to be one day, namely, the time granularity Δt equals one day, $\Delta t = 1$. In reality, for forecasting the options' prices in the next trading day, we always have enough time in the overnight to train the NN.

In the experiments, the option data are chosen from the Hong Kong option market, exemplified as shown in Table 1. The number of options amounts to 320. The data gathered span from date 2005-08-01 to date 2005-10-28 (64 trading days in total). For each day, the data include the stock price, the stock price volatility, and the real option price on the next day. In the calculation, the risk-free interest rate r is assumed 0.025.

It is worth-noting that the data also include the scenario for the deep-in-the money options, exhibited as the very high option prices.

We first simulate the four modified conventional methods (BT, BS, FD, and MC) using the Java APIs on the platform. For the NN forecasting, the Matlab NN toolbox is used, with the Bayesian regularization back-propagation (*trainbr*) adopted as the training algorithm. The reason that we use back-propagation is that it is a proved successful method in forecasting [2]. The number of maximal epochs in training is 100 and the performance goal for the error tolerance is 0.1. For the LNN approach, there is only one node in the output layer with a linear activation function, specifically the *pureline* function. For the multilayer perceptron, a three-layer nonlinear NN is established with *tansig* as the nonlinear activation function.

As aforementioned, the NN acquires the "knowledge" of the current option market via a learning process carried out on the historical data, and then applies this knowledge into further forecasting, in order to gain a better forecasting performance. In appearance, the NN's capability to master the current market's features increases with the size of the training samples, namely a larger L is preferred. However, the empirical studies in the GARCH model imply the length of sliding window should not be very large. In addition, the current research accomplishments in the financial society have demonstrated that the financial data can be considered stationary only during a short span of time. As a consequence, we only experiment with values of L as 3, 5, 8 and 12 in our studies, respectively.

It is well-known that the three-layer feedforward NNs are able to approximate any continuous functions arbitrarily well [4], or the three-layer NN is capable of meeting our needs. Since the Bayesian regularization back-propagation training algorithm (*trainbr*) in the Matlab NN toolbox not only minimizes the errors incurred by training samples, but also those by the magnitudes of weights and biases, which is equivalent to incorporating a penalty function to keep the number of weights to the minimum, the number of the hidden nodes is set to 12 without worrying about overfitting.

In order to make a comparison among different methods regarding the forecast performance, the assessing criterions for the forecasting errors should be given. There are two approaches to define the assessing criterions, the absolute assessing criterions and the relative assessing criterions.

For the absolute assessing criterions, there are a couple of ways to define the average absolute forecasting error. In this paper, we simply use $e = \frac{1}{K} \sum_{t=-K+1}^{0} \left| \hat{C}(t) - C(t) \right|$ for a certain option where K is the number of days that we are concerned. Thus the total average absolute forecasting error for N options is $\bar{e} = \frac{\sum_{j=1}^{N} e_j}{N}$, where e_j represents the forecasting error for the option j. The relative assessing criterion defines the average relative forecasting error in the past T days as $\alpha = \frac{1}{K} \sum_{t=-K+1}^{0} \frac{\left| \hat{C}(t) - C(t) \right|}{C(t-1)}$, for a certain option. Thus the total average relative forecasting error for N options is $\bar{\alpha} = \frac{\sum_{j=1}^{N} \alpha_j}{N}$, where α_j represents the error for the option j. If α or $\bar{\alpha}$ is less than a certain value, say, 10%, the forecast can be considered as an effective one.

In the experiments, we choose $K = 52$ and $N = 320$.

We use the modified conventional methods and the NN approaches to forecast the prices for 320 options, each of whose data spans across three months from 2005-08-01 to 2005-10-28, or 64 trading days. Since some data have to be used in finding the parameters in the modified models, e.g., the volatilities σ, the forecasting work begins from the 12th trading day, and lasts for 52 days.

Fig. 1 shows the total average relative forecasting errors for all the 320 options given by different methods. It can be observed that the NN approaches achieve the errors around 11% which are much better than those around 22% and 23% given by the conventional methods.

It is suggested three points. First, the MLP is likely to be more capable to seize the complex nonlinear features in the option markets. Second, the nonlinearity of the option prices have been largely assimilated by the platform of the four conventional methods, and the NN is able to focus on decreasing the errors. Third, there are still some residual errors on the platform, which can be coped with by the MLP.

For estimating the whole picture of improvement levels across the 320 options, we use the BS method as a base for comparison (it is apparent from the above experiments that the four conventional methods commit the similar levels of errors) and implement a statistic for the distribution of the error decreasing percentages contributed by the MLP, a slightly better NN in empirical studies. In particular, for each option, label e_{MLP} the average absolute forecasting error of the MLP, e_{BS} that of the BS. Define the error decreasing percentage as $\eta = 1 - \frac{e_{\mathrm{MLP}}}{e_{\mathrm{BS}}}$. The distribution of η is shown in Fig. 2. Clearly, $\eta < 1$. If $\eta < 0$, $e_{\mathrm{BS}} < e_{\mathrm{MLP}}$. The larger is η, the more improvement percentage is.

From Fig. 2, it is found that for some 12.5% of options, the MLP does not enhance the forecasting performance, while the average absolute forecasting errors for the other 87.5% of options, the MLP approach is superior to the conventional methods. Additionally, around 10% of the options have achieved an increase in performance more than 0.8. This is equivalent to the fact that for these options, the errors made by the MLP approach are only around 1/5 those of the conventional methods.

An NN can have different lengths of sliding windows. Fig. 3 shows the total average absolute forecasting errors by the LNN and the MLP with different sliding window lengths. We also conduct various experiments based on other values of L, and it is perceived that the adequate lengths of sliding windows, e.g., $L = 12$, produce the best performance (see Fig. 3).

Table 1. The examples of options chosen from the Hong Kong option market (the source of the options' prices is available on the website of the Hong Kong Exchange and Clearing Limited with http://www.hkex.com.hk/tod/markinfo/setdata.asp while the underlying assets' the website of Yahoo finance with http://finance.yahoo.com/q/hp)

option category	option code	option name
BEA	0023.HK	Bank of East Asia
CKH	0001.HK	Cheung Kong
HEH	0006.HK	HK Electric
HWL	0013.HK	Hutchison

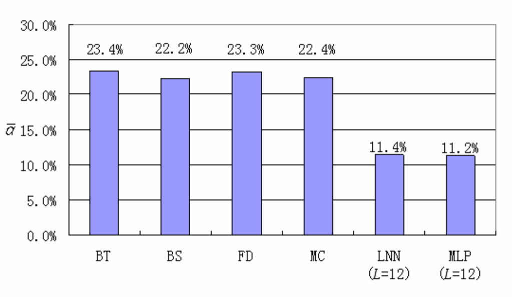


Fig. 1. The average forecasting errors for all options given by the different methods

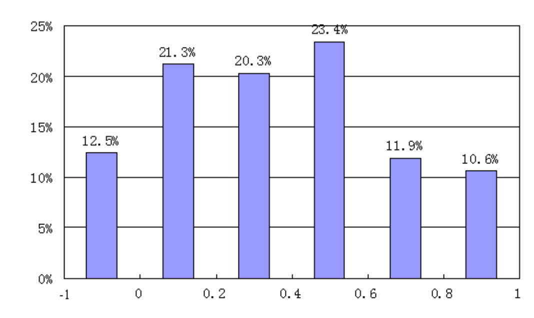


Fig. 2. The distribution of error decreasing percentage η

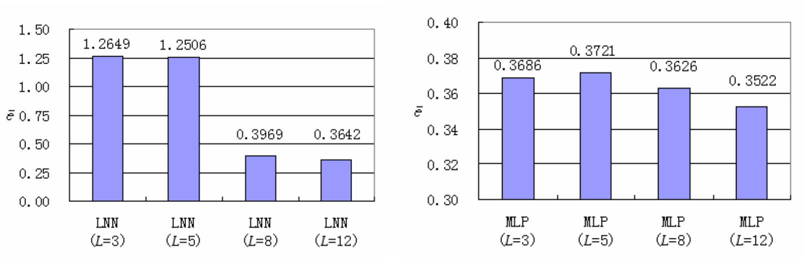


Fig. 3. The total average absolute forecasting errors by the LNN and MLP approaches using different sliding window lengths

5 Conclusions

In this paper, the experimental studies substantiate that in general the NN approaches significantly decrease the forecasted errors compared to the conventional ones, and improve the forecast accuracy accordingly. In addition, the MLP approach outplays the LNN approach slightly. It is positive to believe that the MLP is slightly preferable to the LNN, and the platform upheld by the conventional methods works with a function for shielding the nonlinearity of the option price movements and automatically managing some particular scenarios in option prices. Compared to the conventional option price forecasting methods, the NNs are characteristic of higher forecast accuracy, which enables it a preferable alternative for investors who seek a more effectual approach to seize the option prices.

Apparently, while the option pricing provides the theoretical result and is only instructive to the market practice indirectly, the less-error-incurring NN-based forecasting commits more profits directly for the market practitioners, and enables them to gain the wealth in the option markets realistically.

Future work includes conducting experimental studies with comparisons to other neural network methods as well as other machine learning techniques.

Acknowledgment. The project is sponsored by the NSF of China under grant numbers 70571003 and 70871001.

References

1. Black, F., Scholes, M.: The pricing of options and corporate liabilities. J. Political Economy 81, 637–654 (1973)
2. Lajbcygier, P.: Improving option pricing with the product constrained hybrid neural network. IEEE Trans. Neural Networks 15, 465–476 (2004)
3. Liang, X., Zhang, H., Yang, J.: Pricing options in Hong Kong market based on neural networks. In: King, I., et al. (eds.) ICONIP 2006. LNCS, vol. 4234, pp. 410–419. Springer, Heidelberg (2006)
4. Zurada, J.M.: Introduction to Artificial Neural Systems. West Publishing Company, New York (1992)

High Speed *k*-Winner-Take-ALL Competitive Learning in Reconfigurable Hardware

Hui-Ya Li[1], Yao-Jung Yeh[2], Wen-Jyi Hwang[1,⋆], and Cheng-Tsun Yang[1]

[1] Department of Computer Science and Information Engineering,
National Taiwan Normal University, Taipei, 117, Taiwan
royalfay@gmail.com, whwang@ntnu.edu.tw, alan7486@gmail.com

[2] Networks and Multimedia Institute,
Institute for Information Industry, Taipei, 105, Taiwan
spmark@nmi.iii.org.tw

Abstract. A novel hardware architecture of the competitive learning (CL) algorithm with k-winners-take-all activation is presented in this paper. It is used as a custom logic block in the arithmetic logic unit (ALU) of the softcore NIOS processor for CL training. Both the partial distance search (PDS) module and hardware divider adopt finite precision calculation for area cost reduction at the expense of slight degradation in training performance. The PDS module also employs subspace search and multiple-coefficient accumulation techniques for effective reduction of the computation latency for the PDS search. Experiment results show that the CPU time is lower than that of Pentium IV processors running the CL training program without the support of custom hardware.

1 Introduction

Cluster analysis is a method for partitioning a data set into classes of similar individuals. The clustering applications in various areas such as signal compression, data mining, and pattern recognition, etc., are well documented. In artificial neural networks (ANNs)[4], clustering is commonly realized by competitive learning (CL) techniques [6], where neurons of the network compete among themselves to be activated or fired. The goal of CL is the minimization of the error in clustering analysis/pattern classification [7] or the quantization distortion in vector quantization (VQ) [2].

The simplest CL algorithm is based on the winner-take-all (WTA) [4] scheme, where adaptation is restricted to the single neuron best matching the input pattern. A major problem with the simple WTA learning is the possible existence of dead nodes or the so called under-utilization problem. In such cases, some neurons, due to inappropriate initialization can never become a winner. Therefore, they have no contribution to learning, and may degrade the CL performance.

One effective approach to solve the under-utilization problem is to replace the WTA by the k-winner-take-all (k-WTA) activation [10]. It decreases the

[⋆] To whom all correspondence should be sent.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 594–603, 2009.

dependency on the initial location of the weight vector associated with each neuron. However, since the sorting operations are required for discovering k best matching neurons, the computational cost of CL with k-WTA activation is higher than that of the CL with simple WTA, which already have long computational time for training. Consequently, the k-WTA activation may still not be well-suited for CL clustering demanding fast training.

The drawback of high computational cost may be circumvented by speeding up the search process, e.g., by the employment of partial distance search (PDS) [5] in the original or transform domains, which remove undesired neurons with less number of multiplications. It contains a simple modification to the distortion computation. During the calculation of the distance sum, if the partial distance exceeds the distance to the current best matching neurons, the calculation is aborted. A moderate acceleration usually is achieved by these software approaches.

As compared with the software counterparts, the systolic architectures [9] provide higher throughput. However, these hardware designs may offer limited flexibility. When applied to competitive networks, most systolic architectures are optimized only for a fixed weight vector dimension, and/or a fixed number of neurons. In addition, only the best matching neurons can be identified. These architectures therefore can only be used for CL with WTA activation. Moreover, these architectures do not provide functions for neuron updating, which is important to CL training.

The objective of this paper is to present a novel architecture for the fast training of CL with k-WTA activations. The architecture is based on field programmable gate array (FPGA) [3] so that it is reconfigurable for various competitive networks. FPGA-based reconfigurable hardware can be programmed almost like software, maintaining the most attractive advantage of flexibility with less cost than traditional application specific integrated circuit (ASIC) hardware implementations.

In addition to the employment of FPGAs, a subspace PDS with finite precision calculation in the wavelet domain [8] is adopted to attain high throughput, high flexibility, and low area complexity. In our implementation, coefficients in some highpass subbands will be truncated before fast search because they contain little energy of the input pattern. Moreover, bitplanes of the least significant bits (LSBs) of the remaining coefficients can be removed because they have only a limited impact on the results of the PDS.

To implement the neuron updating process by hardware, a novel circuit for finite precision division is proposed for the computation of learning rate. It is based on simple table lookup process to accelerate the neuron updating process and lower the area cost at the expense of slight degradation in training process.

The proposed implementation has been adopted as a custom logic block in the arithmetic logic unit (ALU) of the softcore NIOS processor [1] running at 50 MHz. The custom instructions are also derived for accessing the custom logic block. The CPU time of the NIOS processor executing the PDS program for k-WTA activation with the custom instructions is measured. Experiment results show that the CPU time is lower than that of 3.0-GHz Pentium processors without the support of custom hardware.

2 Preliminaries

Let $\mathbf{X}$ be the n-stage discrete wavelet transform (DWT) of a $2^n \times 2^n$ vector $\mathbf{x}$ (i.e., the dimension of $\mathbf{x}$ is $2^n \times 2^n$). Then the DWT $\mathbf{X}$ is also a $2^n \times 2^n$ block containing blocks $\mathbf{x}_{L0}$ and $\mathbf{x}_{Vi}, \mathbf{x}_{Hi}, \mathbf{x}_{Di}, i = 0, .., n-1$ [8]. Note that, in the DWT, the blocks $\mathbf{x}_{L(m-1)}$ (lowpass blocks), and $\mathbf{x}_{V(m-1)}, \mathbf{x}_{H(m-1)}, \mathbf{x}_{D(m-1)}$ (V, H and D orientation selective highpass blocks), $m = 1, .., n$, are obtained recursively from $\mathbf{x}_{Lm}$ with $\mathbf{x}_{Ln} = \mathbf{x}$, where the blocks $\mathbf{x}_{Lm}, \mathbf{x}_{Vm}, \mathbf{x}_{Hm}$ and $\mathbf{x}_{Dm}, m = 0, .., n-1$ are with dimension $2^m \times 2^m$, respectively. Therefore, the blocks $\mathbf{x}_{L0}$ and $\mathbf{x}_{Vi}, \mathbf{x}_{Hi}, \mathbf{x}_{Di}, i = 0, .., m-1$, also form the m-stage DWT of the lowpass subband $\mathbf{x}_{Lm}$, denoted by $\mathbf{X}_{Lm}$. That is, $\mathbf{X}_{Lm}$ contains the DWT coefficients of $\mathbf{x}_{Lm}$. The decomposition of $\mathbf{x}_{Lm}$ into four blocks $\mathbf{x}_{L(m-1)}, \mathbf{x}_{V(m-1)}, \mathbf{x}_{H(m-1)}, \mathbf{x}_{D(m-1)}$ can be carried out using a simple quadrature mirror filter (QMF) scheme [8].

Let $\mathbf{y}^j, j = 1, .., N$, be the weight vector associated with j-th neuron. Recall that in the CL algorithm with WTA activation, given a training vector $\mathbf{x}$, the weight vector $\mathbf{y}^{j*}$ satisfying

$$j* = \arg \min_{1 \le j \le N} D(\mathbf{x}, \mathbf{y}^j) \tag{1}$$

will be updated, where $D(\mathbf{u}, \mathbf{v})$ is the squared distance between $\mathbf{u}$ and $\mathbf{v}$, and both vectors have the same dimension $2^n \times 2^n$. The updated y^{j*} is then given as

$$\mathbf{y}^{j*} \leftarrow \mathbf{y}^{j*} + \eta^{j*}(\mathbf{x} - \mathbf{y}^{j*}), \tag{2}$$

where η^{j*} is the learning rate of the $j*$-th neural unit.

In wavelet-based PDS algorithm, the DWT of the training data and initial weight vectors are given before the design. Let $\mathbf{X}$ and $\mathbf{Y}^j$ be the DWT of $\mathbf{x}$ and $\mathbf{y}^j$, respectively. Let X_i and Y_i^j be the i-th coefficient of $\mathbf{X}$ and $\mathbf{Y}^j$ in the zig-zag order, respectively. Moreover, let $D^q(\mathbf{X}, \mathbf{Y}^j) = \sum_{i=1}^{q}(X_i - Y_i^j)^2$ be the partial distance between $\mathbf{X}$ and $\mathbf{Y}^j$. Since $D(\mathbf{X}, \mathbf{Y}^j) > D^q(\mathbf{X}, \mathbf{Y}^j)$, it can be proved that

$$D(\mathbf{x}, \mathbf{y}^j) > D^q(\mathbf{X}, \mathbf{Y}^j). \tag{3}$$

The PDS for finding the optimal weight vector $\mathbf{y}^{j*}$ proceeds as follows. Given a training vector $\mathbf{x}$, first we initialize the current best-matching vector to be $\mathbf{y}^p$, where $p = \arg\min_j D^1(\mathbf{X}, \mathbf{Y}^j)$, and the current minimum distortion D_{min} to be $D(\mathbf{x}, \mathbf{y}^p)$. Starting from $\mathbf{Y}^1$, we check each vector (except initial *current* $\mathbf{Y}^p$ itself) until $\mathbf{Y}^N$ is reached. Suppose $\mathbf{Y}^j$ is the current vector to be searched. Beginning with $q = 1$, for each value of q, $q = 1, .., 2^n \times 2^n$, we evaluate $D^q(\mathbf{X}, \mathbf{Y}^j)$. If $D^q(\mathbf{X}, \mathbf{Y}^j) > D_{min}$, then it also follows from eq.(3) that $D(\mathbf{x}, \mathbf{y}^j) > D_{min}$ and $\mathbf{Y}^j$ can be rejected. Otherwise, we go to the next value of q and repeat the same process. Note that the partial distance $D^q(\mathbf{X}, \mathbf{Y}^j)$ can be expressed as

$$D^q(\mathbf{X}, \mathbf{Y}^j) = D^{q-1}(\mathbf{X}, \mathbf{Y}^j) + (X_q - Y_q^j)^2. \tag{4}$$

Therefore, the partial distance of the new q can use the partial distance of the previous q, and only the computation of $(X_q - Y_q^j)^2$ is necessary.

This PDS process is continued until $\mathbf{Y}^j$ is rejected or q reaches $2^n \times 2^n$. If $q = 2^n \times 2^n$, then we compare $D(\mathbf{x}, \mathbf{y}^j)$ with D_{min}. If $D(\mathbf{x}, \mathbf{y}^j) < D_{min}$, then the current minimum distortion D_{min} is replaced by $D(\mathbf{X}, \mathbf{Y}^j)$ and the current best-matching vector to $\mathbf{x}$ is set to $\mathbf{y}^j$. After all the weight vectors are searched, the final current best-matching vector is then the $\mathbf{y}^{j*}$, and the D_{min} is the corresponding distance. After $\mathbf{y}^{j*}$ is found, we update $\mathbf{Y}^{j*}$ by

$$\mathbf{Y}^{j*} \leftarrow \mathbf{Y}^{j*} + \eta^{j*}(\mathbf{X} - \mathbf{Y}^{j*}). \tag{5}$$

Note that, since DWT is linear, the updated $\mathbf{Y}^{j*}$ in the left hand side (LHS) of eq.(5) is the DWT of the updated $\mathbf{y}^{j*}$ in the LHS of eq.(3).

3 The FPGA Implementation

3.1 PDS for Hardware Realization

The PDS adopted here for hardware realization features the subspace search, the bitplane reduction, and multiple-coefficient partial distance accumulation.

Subspace search and bitplane reduction: Because the DWT is able to compact the energy of a vector to a lowpass subband, the PDS in the wavelet domain can be accelerated further by scanning only the coefficients in the lowpass subband. The weight vector associated with each neuron for subspace search is $\mathbf{Y}_{Lm}^j$(i.e., the DWT coefficients of $\mathbf{y}_{Lm}^j$), $j = 1, ..., N$. Therefore, the subspace search is also beneficial for the VLSI realization since it significantly reduces the storage size of weight vectors.

Although the $\mathbf{Y}_{Lm}^j, j = 1, ..., t$, can be obtained offline, the $\mathbf{X}_{Lm}$ should be obtained online from a source vector $\mathbf{x}$. Therefore, the DWT computation is still necessary in the subspace PDS. We use the Haar wavelet for the DWT hardware realization because it has simple lowpass filter (i.e., impulse response $= \{\frac{1}{2}, \frac{1}{2}\}$) and highpass filer (i.e., impulse response $= \{-\frac{1}{2}, \frac{1}{2}\}$). Consequently, no multiplication is necessary for the DWT implementation.

In addition to subspace search, the finite precision calculation can further reduce the area cost for PDS operations. Although the hardware implementation of the floating point PDS is possible, the corresponding area cost for storing $\mathbf{Y}_{Lm}^j$, $j = 1, ..., N$, may be high. In the finite precision calculation technique, all coefficients in $\mathbf{X}_{Lm}$ and $\mathbf{Y}_{Lm}^j$, $j = 1, ..., N$, are represented by b bitplanes. This representation is also beneficial for lowering the area complexity of squared distance calculation and sorting circuit.

Multiple-coefficient partial distance accumulation: From eq.(4), it follows that the partial distance is accumulated one coefficient at a time in the basic PDS. Therefore, the hardware realization of the basic PDS requires only one multiplier. The speed of partial distance computation can be accelerated by accumulating δ coefficients at a time. That is, δ multipliers are employed for computing the squared distance concurrently in hardware. In the subspace PDS, the partial distance is then computed by

$$D^q(\mathbf{X}_{Lm}, \mathbf{Y}^j_{Lm}) = D^{q-\delta}(\mathbf{X}_{Lm}, \mathbf{Y}^j_{Lm}) + \sum_{i=q-\delta+1}^{q} (X_i - Y^j_i)^2, \qquad (6)$$

where X_i and Y^j_i are the i-th coefficient of $\mathbf{X}_{Lm}$ and $\mathbf{Y}^j_{Lm}$, respectively. These coefficients are indexed in the zig-zag order. The size of the lowpass subband $2^m \times 2^m$ should be a multiple of δ.

In our design, the vector squared distance computation (VSDC) unit is responsible for the computation of $\sum_{i=q-\delta+1}^{q}(X_i - Y^j_i)^2$ on the right hand side of eq.(6). The VSDC contains δ scalar squared distance computation (SSDC) unit, which computes the squared distance between two coefficients. An adder tree then is employed to collect the results of all the δ SSDC units.

3.2 Sorting Operations for k-WTA

For the CL with simple WTA activation, only the best matching neuron is identified. No sorting circuit is required because the PDS can be used for finding the winning neuron. However, k-WTA activation is adopted in our implementation. It is thus necessary to employ a sorting circuit for the discovery of k best matching neurons.

To illustrate this fact in more detail, we first let $J = \{j_1, j_2, ..., j_k\}$ be the set of indices of k best matching neurons to an input vector $\mathbf{x}$ such that $D(\mathbf{X}_{Lm}, \mathbf{Y}^{j_1}_{Lm}) \leq ... \leq D(\mathbf{X}_{Lm}, \mathbf{Y}^{j_k}_{Lm})$ and $D(\mathbf{X}_{Lm}, \mathbf{Y}^{j_k}_{Lm}) \leq D(\mathbf{X}_{Lm}, \mathbf{Y}^{j}_{Lm})$ for any j not belong to J. The goal of the PDS for CL with k-WTA activation is to find the set J, and then update all neurons whose indices belong to J.

Let $D_i = D(\mathbf{X}_{Lm}, \mathbf{Y}^{j_i}_{Lm})$, $i = 1, ..., k$. Suppose the subspace PDS search up to the weight vector $\mathbf{y}^{j-1}$ has been completed, and the search over $\mathbf{y}^j$ is to be done. We then compare the distance $D(\mathbf{X}_{Lm}, \mathbf{Y}^j_{Lm})$ with D_k using the subspace PDS. When $D(\mathbf{X}_{Lm}, \mathbf{Y}^j_{Lm}) < D_k$ the neuron with index j_k is no longer one of the k best matching neurons to the input vector. In this case, we will update J by first removing j_k. The new set J for the next PDS search over $\mathbf{y}^{j+1}$ is formed by inserting j into J, and then sorting the index j and $j_1, j_2, ..., j_{k-1}$ in accordance with their respective distances to the input vector. This process is repeated until the PDS over the weight vector of the final neuron(i.e., $\mathbf{y}^N$) is completed. The resulting J then contains the k best matching neurons to the input vector.

3.3 Neuron Updating

After the discovery of J, the proposed architecture then updates all the weight vectors having index in J. Let $j*$ be an index in J, the corresponding weight vector is then updated using eq.(5). The learning rate need to be considered in the updating process. In our implementation, we set

$$\eta^{j*} = \frac{1}{4 \times r}, \qquad (7)$$

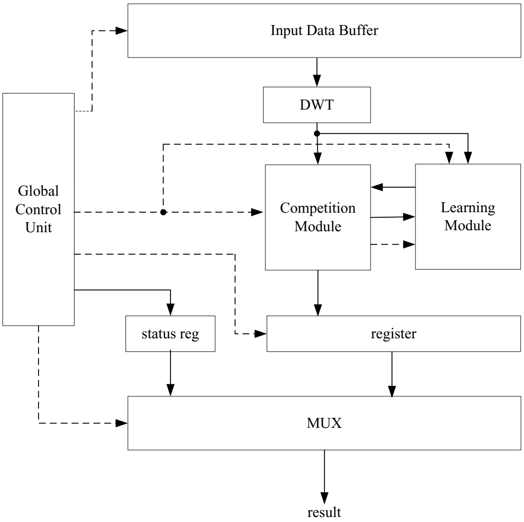


Fig. 1. The basic VLSI architecture for realizing the proposed CL algorithm

where r denotes current number of times the weight vector $\mathbf{y}^{j*}$ has been selected for updating. The computation of learning rate involves division. In our design, a lookup table based divider is adopted for reducing the area complexity and accelerating the updating process. Given any integer $w > 0$, eq.(7) can then be rewritten as

$$\eta^{j*} = \frac{2^w}{r} \times 2^{-(w+2)}. \tag{8}$$

Consequently, the architecture for learning rate computation contains 3 components: counter, ROM and shift circuit. The counter is used for recording the number of times $\mathbf{y}^{j*}$ has been selected for updating. Based on the output of of the counter, r, the ROM is used for obtaining the finite precision value of $\frac{2^w}{r}$. The shift circuit then shift the output of ROM left by $(w + 2)$ positions.

3.4 The CL Architecture

The basic VLSI architecture for realizing the CL with k-WTA is shown in Fig. 1, which contains a DWT unit, a competition module, a learning module, a global control unit, and a number of registers. The DWT unit is used for computing $\mathbf{X}_{Lm}$ of an input vector $\mathbf{x}$. The competition module performs subspace PDS for identifying k winning neurons. Based on the search result provided by the competition module, the learning module then updates the weight vectors associated with the winning neurons.

Fig. 2 shows the architecture of competition module. This module performs subspace PDS and continuous sorting operations for the discovery of J. The input vector $\mathbf{X}_{Lm}$ is obtained from the DWT unit. Weight vectors $\mathbf{Y}^j_{Lm}, j = 1, ..., N$, are provided by the learning module. The VSDC and accumulator in

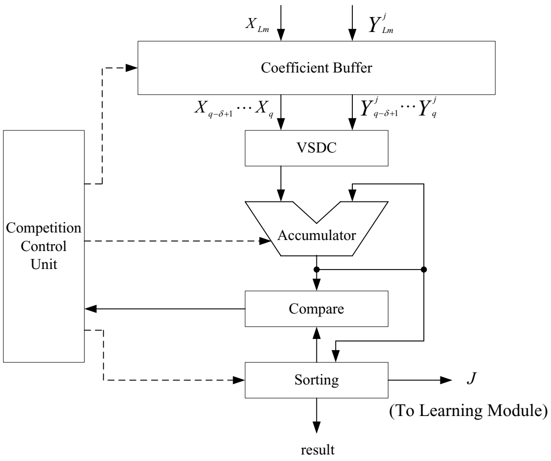


Fig. 2. The architecture of competition module

the module are used to compute and store the partial distance $D^q(\mathbf{X}_{Lm}, \mathbf{Y}^j_{Lm})$ for $q = \delta, 2\delta, ..., 2^m \times 2^m$. The comparator then compares $D^q(\mathbf{X}_{Lm}, \mathbf{Y}^j_{Lm})$ with D_k. Comparison results are reported to the control unit.

When $D^q(\mathbf{X}_{Lm}, \mathbf{Y}^j_{Lm}) \leq D_k$, q is incremented by δ, and the PDS based on the VSDC and accumulator continues. Moreover, in case q reaches $2^m \times 2^m$, and $D^q(\mathbf{X}_{Lm}, \mathbf{Y}^j_{Lm}) \leq D_k$, the index j_k is removed from J, and the index j of the current weight vector becomes new member of J. The sorting circuit is then activated for re-ordering the elements of J. On the other hand, when $D^q(\mathbf{X}_{Lm}, \mathbf{Y}^j_{Lm}) > D_k$, the PDS for current weight vector is aborted. The PDS for the next weight vector then is activated by the controller.

Fig. 3 reveals the architecture of learning module. The RAM stored in the module contains all the weight vectors in the neural network. There are N counters in the module. Counter i stores integer r (as shown in eq.([8])) for the computation of current learning rate associated with the i-th neuron. The r associated with each weight vector to be updated in J is used as the index for table lookup. The output of the table is then the learning rate for the update unit containing subtractor, multiplier, and adder. The weight vectors having index in J are updated one at a time so that only one table is required for computing the learning rate. The updated weight vector at the output of the update unit is then delivered back to the RAM in the learning module.

4 Experimental Results and Concluding Remarks

Table 1 shows the area complexity of the VSDC units with various combination of δ (number of SSDC units) and b (number of bitplanes). The FPGA used for the area complexity measurement is the Altera Stratix EP1S40. It can be observed from the table that, although VSDC units with larger δ values have

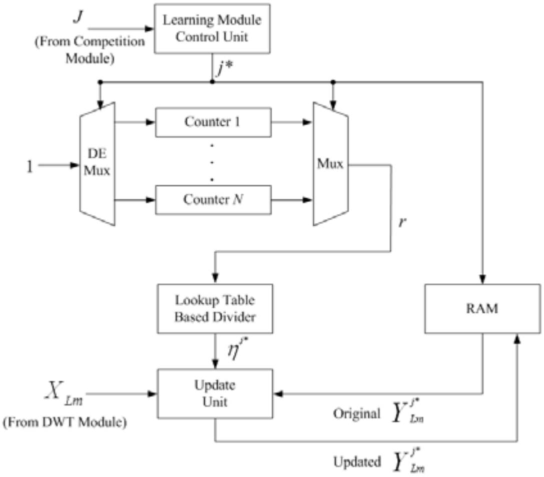


Fig. 3. The architecture of learning module

Table 1. The area complexity of the VSDC units with various combinations of δ (number of SSDC units) and b (number of bitplanes)

		Number of Bitplanes b			
		14	12	10	8
	1	46	40	34	28
Number of	2	123	107	92	75
SSDC units	4	278	242	206	170
δ	8	589	513	437	361
	16	1212	1056	900	744

higher throughput, their area complexity is also very large. In particular, when $\delta = 16$ and $b = 14$, the area complexity is 1212 logic elements (LEs). Therefore, the employment of VSDC with very large δ values may be difficult for some applications where the cost is an important concern. On the contrary, smaller δ values (e.g., $\delta = 4$) requires significantly lower area complexity while improving the throughput. Using $b = 8$, the VSDC with $\delta = 4$ uses only 170 LEs, which is only 14.02 % of the LEs consumed by VSDC with $\delta = 16$ and $b = 14$.

Table 2 investigates the impact of b on the memory bits consumption and image distortion. We use the peak SNR (PSNR) as the performance measure in the table. The PSNR is defined as $10 \log \frac{255^2}{\bar{D}}$, where $\bar{D}$ is defined as the mean squared error between an original pixel and its reconstruction in an image. Both the 512×512 images "Lena" and "Zelda" are used for the training. The images "Girl" "Baby" and "House" are then used as the test images for the performance measurement. We set $N = 256$ and $k = 5$ for the k-WTA. The vector dimension is 8×8 (i.e., $n = 3$). When the subspace search is adopted, the dimension of subvectors for the PDS is 4×4 (i.e., $m = 2$).

Table 2. The PSNR values and memory bit consumption of neuron sets for various number of bitplanes b with or without subspace search

Subspace		No	Yes			
b		14	14	12	10	8
PSNR (dB)	Girl	30.2219	30.0063	30.0036	29.9647	29.8658
	Baby	26.9533	26.6825	26.6782	26.6729	26.5793
	House	25.9510	25.6674	25.6667	25.6648	25.6115
Memory Bits		131072	61440	53248	45056	36864

Table 3. The PSNR values of CL training based on different dividers

Dividers	House	Lena	Tree
Long Division	26.3675	24.2825	21.5550
Table Lookup	26.3656	24.2814	21.5478

From Table 2, it can be observed that the PSNR values are only slightly degraded when subspace search is adopted and/or b values are lowered. In particular, when the test image is “Girl,” the PSNR value is only degraded by 0.3561 dB (from 30.2219 dB to 29.8658 dB) by the employment of subspace search with $b = 8$. Moreover, the memory bits consumption for subspace search with $b = 8$ is only 36864 bits. By contrast, the memory bits consumption for PDS without subspace search is 131072 bits. The reduction in storage size with subspace search and bitplane reduction ($b = 8$) is therefore 71.88 %.

Our proposed table lookup based divider can simplify the neuron updating process at the expense of slight performance degradation. As shown in Table 3, as compared with the usual floating point divider based on long division, the degradation in PSNR using our divider is less than 0.01 dB for all the test images “House”, “Lena” and “Tree.” In this experiment, the number of weight vectors is $N = 64$. Two 512×512 images “Baboo” and “Bridge” are used for the training.

Finally, Table 4 compares the execution time of NIOS [1] with that of the Pentium IV for various PDS operations, where the execution time is defined as the average CPU time (in μSec) required for identifying the k best matching neurons and updating these neurons to each input vector. The NIOS CPU running with the support of the proposed hardware is considered in the table; whereas, the Pentium IV CPU is executing solely on C codes. Note that, in the NIOS system, our architecture is used as a custom logic in the ALU. It can be accessed by custom instructions. In the experiments, the number of weight vectors is $N = 1024$. Dimension of vectors is 8×8 ($n = 3$). We set $k = 5$ for the k-WTA activation. Based on the numerical results shown in Tables 1 and 2, the subspace PDS operates with $m = 2, b = 8$ and $\delta = 4$. The whole NIOS system (consisting of NIOS CPU and proposed circuit) consumes 27668 LEs. The system therefore uses only 65.64 % of the LEs of the target FPGA device.

It can be observed from Table 4 that the average CPU time of the 50 MHz NIOS systems with the aid of the proposed circuit is lower than that of the 3.0

Table 4. The CPU time of various CL systems

CPU type	Pentium IV 3.0GHz		NIOS II 50MHz
Algorithm	Full Search	Subspace+PDS	Subspace+PDS
Implementation	Software	Software	Hardware/Software Codesign
CPU Time(μs)	20877.41	350.93	86.19

GHz Pentium IV systems executing the same subspace PDS algorithm. Table 4 also includes the CPU time of the CL with k-WTA executed in Pentium IV without subspace search. In this case, the CPU time is 20877.41 μS for Pentium IV 3.0-GHz CPU, which is 242.22 times longer then the NIOS CPU with the proposed hardware support. All these facts demonstrate the effectiveness of the proposed CL architecture.

References

1. NIOS II Processor Reference Handbook (2008), Altera Corporation, http://www.altera.com/literature/lit-nio2.jsp
2. Gersho, A., Gray, R.M.: Vector Quantization and Signal Compression. Kluwer, Norwood (1992)
3. Hauck, S., Dehon, A.: Reconfigurable Computing. Morgan Kaufmann, San Francisco (2008)
4. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall, Englewood Cliffs (1998)
5. Hwang, W.J., Lin, F.J., Zeng, Y.C.: Fast Design Algorithm for Competitive Learning. Electronics Letters, 1469–1470 (1997)
6. Kohonen, T.: The self-organizing map, 3rd edn. Springer, Heidelberg (2000)
7. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 3rd edn. Academic Press, London (2006)
8. Walnut, D.F.: An Introduction to Wavelet Analysis. Birkhauser, Basel (2002)
9. Wang, C.L., Chen, L.M.: A New VLSI Architecture for Full-Search Vector Quantization. IEEE Trans. Circuits and Systems for Video Technology, 389–398 (1996)
10. Yen, J.C., Guo, J.I., Chen, H.C.: A new k-winners-take-all neural network and its array architecture. IEEE Trans. Neural Networks, 901–912 (1998)

Adaptive Neurofuzzy Network Based PI Controllers with Multi-objective Functions

Y.F. Chan, C.W. Chan, and H.T. Mok

Department of Mechancial Engineering, The University of Hong Kong, Hong Kong, China
flysky@hkusua.hku.hk, mechan@hku.hk

Abstract. As the performance of PI controllers can deteriorate rapidly for highly nonlinear systems, nonlinear PI controllers are developed. An approach to design these controllers is to switch between several linear PI controllers using fuzzy logic based on the Takagi-Sugeno model. Following this approach, nonlinear PI controllers are derived in this paper using B-spline neurofuzzy networks. Design guidelines and on-line training of the proposed controller are devised, and the performance is illustrated by a simulated two-tank water level control rig. Comparison with conventional PI controllers is also made.

Keywords: Nonlinear PI controller, neurofuzzy networks, adaptive control.

1 Introduction

The proportional-integral (PI) controllers are still popular in industrial applications for their ease of installation and robustness. Many methods for designing PI controllers with constant parameters are available [1]. However, the performance of constant parameter PI controllers for nonlinear systems can deteriorate rapidly. To overcome this problem, self-tuning PI controllers are proposed [2], as they can adapt to the nonlinearity of the process. However, their performance may not always be satisfactory during the adaptation process. Further, the controller parameters have to be re-tuned whenever the operating points are changed. Recently, neural networks and fuzzy logic are used increasingly in the design of nonlinear controllers with PID structure [3] and [4]. However, the drawback is that these controllers require a long training time to tune the fuzzy rules, or the weights of the neural networks.

A simple, but effective method to control nonlinear systems is the multi-model controller [5] that switches between linear controllers designed at specific operating points. However, the transition from one local region to another may not always be smooth. As neurofuzzy networks [8] can approximate nonlinear functions with arbitrary accuracy, and that the transition from one local region to another is smooth, they are used to develop self-tuning nonlinear controllers with the local linear controllers derived by the generalized minimum variance sub-optimal control law [6], and the adaptive nonlinear PI controllers [10]. However, these controllers are derived using the same sub-optimal control law, and the performance may not always be satisfactory. In this paper, the adaptive nonlinear PI controller developed in [10] is extended by using a different sub-optimal control law at each of the operation points.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 604–613, 2009.

This paper is organized as follows. The nonlinear system is introduced in Section 2, followed by a brief review of the neurofuzzy networks in Section 3. The neurofuzzy networks based nonlinear PI controller is derived in Section 4, and the training target and the learning algorithm for the proposed controllers in Section 5. The performance of the proposed controller to control a two-tank water level system and comparison with conventional PI controllers is presented in Section 6.

2 Preliminaries

Consider the nonlinear autoregressive-moving average (NARMAX) model [7]:

$$y(t) = f(y(t\text{-}1),\ldots, y(t\text{-} n_y),u(t\text{-}k),\ldots, u(t\text{-}k\text{-} n_u), \zeta(t\text{-}1),\ldots, \zeta(t\text{-} n_\zeta)) + \zeta(t)\ . \tag{1}$$

where $f(.)$ is a smooth nonlinear function, $y(t)$ and $u(t)$ are respectively the system output and the control, $\xi(t)$ is an uncorrelated random noise with zero mean, n_y, n_u, n_ξ and k are known system orders and time delay. As $f(.)$ is smooth, it can be approximated by Taylor series expansion at selected operating points, say at O_t. Ignoring higher order terms, the NARMAX model can be approximated by the following linear ARMAX model [7]:

$$A(O_t)y(t) = z^{-k}B(O_t)u(t) + C(O_t)\zeta(t) + \delta(O_t)\ . \tag{2}$$

where $A(.)$, $B(.)$ and $C(.)$ are polynomials in z^{-1}, the backward shift operator. A general nonlinear controller for the nonlinear system (1) can be expressed as follows [8]:

$$u(t)=g(u(t\text{-}1),\ldots,u(\mathrm{t}\text{-} n_f), y(t),\ldots,y(t\text{-} n_g),w(t),\ldots, w(t\text{-} n_h))\ . \tag{3}$$

where $g(.)$ is a smooth nonlinear function, $w(t)$ is the reference input, and n_f, n_g and n_h are the orders of the controller. Similar to the derivation of (2), the controller (3) can be expressed in a form similar to (2) with appropriate orders. For controller (3) to have a PI structure, it is necessary to set $n_f = 0$, $n_g = n_h = 1$. The design of the nonlinear PI controller involves determining the nonlinear function $g(.)$. Since the nonlinear system (1) is smooth, the nonlinear controller (3) can also be assumed to be smooth. It follows that the nonlinear system (1) can be controlled locally by a linear PI controller, and the nonlinear PI controller (3) can be implemented by incorporating all the local linear PI controllers using the Takagi-Sugeno model.

The design of the nonlinear PI controller (3) using this approach involves two steps. First, linear PI controllers are designed at selected local regions, and second, a mechanism is devised to integrate the local linear PI controllers. In this paper, it is proposed that the local linear PI controllers are designed based on the sub-optimal control law for deriving the self-tuning integrating controller [9], and they are being integrated to form the nonlinear PI controllers using neurofuzzy networks, which are shown to be an implementation of the Takagi-Sugeno model [8]. A brief review of neurofuzzy networks and the online training algorithm is presented next.

3 Neurofuzzy Networks

The B-spline neurofuzzy network (*BSNN*) is shown in Fig. 1. As it incorporates fuzzy rules, expert knowledge in linguistic form can also be handled by the network [8].

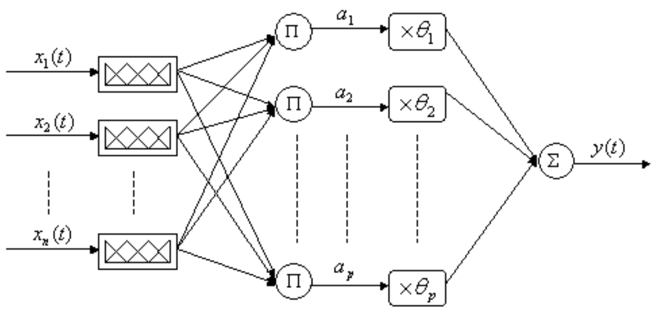


Fig. 1. B-spline neurofuzzy network with triangular basis function as the membership functions

B-spline functions are used here as the membership functions of the fuzzified network input $x(t) = x = [x_1, \ldots, x_n]^T$, since they are compact and are always positive over its support, e.g., $\mu_\rho^j(x)=0,\ x\notin[\sigma_{j-\rho},\sigma_j]$ and $\mu_\rho^j(x)>0$, $x\in(\sigma_{j-\rho},\sigma_j)$, and they form a partition of unity, i.e. $\sum_j \mu_\rho^j(x)\equiv 1$, $x\in[x_{\min},x_{\max}]$ where σ is the knot, and $\mu_\rho^j(x)$ is the j^{th} membership function of order ρ. A ρ^{th} order B-spline function is given by:

$$\mu_\rho^j(x) = \left(\frac{x-\sigma_{j-\rho}}{\sigma_{j-1}-\sigma_{j-\rho}}\right)\mu_{\rho-1}^{j-1}(x) + \left(\frac{\sigma_j - x}{\sigma_j-\sigma_{j-\rho+1}}\right)\mu_{\rho-1}^{j}(x) \quad . \qquad (4)$$
$$\mu_1^j(x) = \begin{cases} 1 & if\ x\in[\sigma_{j-1},\sigma_j) \\ 0 & otherwise \end{cases}$$

Note that a triangular basis function is one with an order of two. Consider a fuzzy rule R_p in the *BSNN*:

R_p: IF $x_1(t)$ is negative big, and $x_2(t)$ is positive medium, and …, and $x_n(t)$ is negative small, THEN $y(t)$ is negative medium.

The output of the *BSNN*, $y(t)$ = *BSNN*($x(t)$), obtained from the rules of the network is a function of the input x. If the fuzzy input set is a singleton, *T*-norm and *S*-norm operators are used to implement respectively the product and sum functions in the fuzzy operation, and the centre-of-gravity defuzzification algorithm is used, $y(t)$ is a linear function of the transformed input, $y(t)$ = *BSNN*($x(t)$)=$a^T(x(t))\theta$, where θ is the network weights, $a(x(t)) = [a_1(x(t)), a_2(x(t)), \ldots, a_p(x(t))]$, $a_i(x(t))$ is the tensor product of the univariate B-spline basis functions $\mu_\rho^j(x)$, i.e. $a_i(x(t)) = \prod_{k=1}^{n}\mu_\rho^j(x_k(t))$, and p is the number of weights in the network given by,

$$p=\prod_{i=1}^{n}(r_i+\rho_i) \quad . \qquad (5)$$

ρ and r are respective the order of the basis functions and the number of inner knots in the partition of $x(t)$.

4 Nonlinear PI Controllers

The PI controller in continuous time is given by,

$$u(t) = K\left[e(t) + \frac{1}{T_i}\int e(t)dt \right] . \tag{6}$$

where K is the proportional gain, and T_i, the integral time constant, $e(t) = w(t) - y(t)$, is the error. A velocity form of the discrete PI controller is,

$$\Delta u(t) = u(t) - u(t\text{-}1) = k_0[w(t)\text{-}y(t)] + k_1[w(t\text{-}1)\text{-}y(t\text{-}1)] . \tag{7}$$

with $k_0=K$ and $k_1= K(-1+T/T_i)$, where T is the sampling interval. The integrating control law (A3) derived in the Appendix can be rewritten as,

$$\Delta u(t) = \frac{-H(z^{-1})}{F(z^{-1})}\left[w(t) + \frac{G(z^{-1})}{H(z^{-1})} y(t) \right] . \tag{8}$$

Following [2], the system is assumed to be approximated by a first order model with $C(z^{-1}) = 1$,

$$(1\text{-}\ a_1 z^{-1})y(t) = z^{-1} b_0\, u(t) + \xi(\mathrm{t})/\Delta . \tag{9}$$

To obtain the PI structure given by (7), $G(z^{-1})$ and $H(z^{-1})$ are chosen, such that $G(z^{-1}) = -H(z^{-1})$ and the order of $F(z^{-1})$ and $G(z^{-1})$ are respectively 0 and 1 Then (8) becomes,

$$\Delta u(t) = \frac{g_0}{f_0}[w(t) - y(t)] + \frac{g_1}{f_0}[w(t-1) - y(t-1)] . \tag{10}$$

Let $P(z^{-1})$, $Q(z^{-1})$ and $R(z^{-1})$ in the generalized system output (A2) be chosen as,

$$P(z^{-1}) = 1 + \lambda_P z^{-1};\ \ Q(z^{-1}) = \lambda_Q;\ \ R(z^{-1}) = (1 + a_1 + \lambda_P) - a_1 z^{-1} . \tag{11}$$

where λ_P and λ_Q are constants. From (9), (11) and (A3), $F(z^{-1})$, $G(z^{-1})$ and $H(z^{-1})$ are:

$$F(z^{-1}) = b_0 + q_0 ;\ G(z^{-1}) = (1 + a_1 + p_1) - a_1 z^{-1};\ H(z^{-1}) = -(1 + a_1 + p_1) + a_1 z^{-1} . \tag{12}$$

The PI controller (10) becomes,

$$\Delta u(t) = \frac{1 + a_1 + \lambda_P}{b_0 + \lambda_Q}[w(t) - y(t)] + \frac{-a_1}{b_0 + \lambda_Q}[w(t-1) - y(t-1)] . \tag{13}$$

From (7), K and T_i are given by [10],

$$K = \frac{1 + a_1 + \lambda_P}{b_0 + \lambda_Q};\qquad T_i = \frac{1 + a_1 + \lambda_P}{1 + \lambda_P} T . \tag{14}$$

From (14), the gain K is proportional to λ_P, but inversely proportional to λ_Q, whilst the integral time constant T_i is inversely proportional to λ_P. It implies that increasing λ_P increases K, but reduces T_i, whilst increasing λ_Q reduces only K. The design of λ_P and λ_Q is now reduced to choosing the gain and the integral time constant of PI controllers, as in conventional PI tuning methods. However, it should be noted that these are only some useful guidelines for choosing λ_P and λ_Q, as the system is assumed to be approximated by a first order system.

5 Implementation of Nonlinear PI Controllers

The nonlinear PI controller (10) is now implemented by the following *BSNN*,

$$\Delta u(t) = BSNN\,[y(\mathrm{t}),y(t\text{-}1),w(t),\, w(t\text{-}1)]\ . \tag{15}$$

Normalizing (A3) by f_0, the leading constant of $F(z^{-1})$, gives:

$$\Delta u(t) = -\overline{F}(z^{-1})\Delta u(t) - \overline{G}(z^{-1})y(t) - \overline{H}(z^{-1})w(t)\ . \tag{16}$$

Note that $\overline{H}(z^{-1})$ is independent of the parameters of the system, as it is a function of C and R with the former set to 1 and the latter a linear term selected by the user. Consequently, the controller (16) can be implemented using two *BSNN*s, one with $u(t)$ and $y(t)$ as the inputs, and the other with $w(t)$ as the input, as follows,

$$\Delta u(t) = BSNN_1\,[y(\mathrm{t}),y(t\text{-}1)] + BSNN_2\,[w(t),\, w(t\text{-}1)]\ . \tag{17}$$

In regulating a process, it is often that the set-point of the system is a step change, i.e., $w(t) = w(t\text{-}1) =$ constant, except at the time when the step change occurs. Therefore, (17) can be further simplified to,

$$\Delta u(t) = BSNN_1\,[y(\mathrm{t}),y(t\text{-}1)] + BSNN_2\,[w(t)]\ . \tag{18}$$

From (5), the total number of weights in the two *BSNN*s in (18) is much less that in (15), giving a significant saving in computation time. For convenience, the nonlinear PI controller (18) is referred to as the *BSNN-PI* controller in the following discussions.

5.1 Training of *BSNN*-PI Controllers

From (16), the generalized system output $\phi_i(t)$ given by (A2) for the i^{th} operating point can be rewritten as,

$$\phi_i(t) = [\Delta u(t-k) + \overline{G}_i y(t-k) + \overline{H}_i w(t-k)] + E\xi(t)\ . \tag{19}$$

From the above discussion, (19) can be implemented by two *BSNN*s, as follows,

$$\phi_i(t) = \{\Delta u(t\text{-}k) - BSNN_1\,[y(t\text{-}k),y(t\text{-}k\text{-}1)] - BSNN_2\,[w(t\text{-}k)]\ \} + E\zeta(t)\ . \tag{20}$$

Rearranging (20) yields:

$$BSNN_1\,[y(t\text{-}k),y(t\text{-}k\text{-}1)] + BSNN_2\,[w(t\text{-}k)] = [\Delta u(t\text{-}k) - \phi_i(t)] + E\zeta(t)\ . \tag{21}$$

As $\xi(t)$ is a zero mean white noise, $E\xi(t)$ is also zero mean, and uncorrelated with the other terms on the right-hand side of (21), giving the training target, $\psi_i(t)=\Delta u(t-k)-\phi_i(t)$, where $\phi_i(t)$ is computed by (A2). As the output of the *BSNN* is linear in the weights of the network, the on-line estimate of these weights, denoted by $\hat{\theta}(t)$, can be obtained using the well known recursive least squares (RLS) method [9], as follows,

$$\hat{\theta}(t)=\hat{\theta}(t-1)+\frac{P(t-1)a(x(t-k))[\psi_i(t)-a^T(x(t-k))\hat{\theta}(t-1)]}{1+a^T(x(t-k))P(t-1)a(x(t-k))}\ . \tag{22}$$

where the covariance matrix, $P(t)$ is given by:

$$P(t)=P(t-1)-\frac{P(t-1)a(x(t-k))a^T(x(t-k))P(t-1)}{1+a^T(x(t-k))P(t-1)a(x(t-k))}\ . \tag{23}$$

After the weights of the *BSNN*s are updated, the control $\Delta u(t)$ is computed from (18).

5.2 Guidelines for Selecting Design Parameters

The implementation of the nonlinear PI controller (18) involves selecting first $P_i(z^{-1})$, $Q_i(z^{-1})$ and $R_i(z^{-1})$ in the generalized system output for i = 1, …, K, where K is the number of operating points, and then initializes the network weights and covariance matrix in the RLS estimator, and the range, the order of the basis functions and the number of inner knot for the inputs of the *BSNN* using the following guidelines.

(i) From (A2), $P_i(z^{-1})$ and $Q_i(z^{-1})$ are selected by choosing λ_{Pi} and λ_{Qi}. However, the selection of $R_i(z^{-1})$ is more complex, as it involves the system parameter. For unknown systems, it can be chosen based on the steady state errors. The closed-loop output of the local linear system with the linear PI control (13) is,

$$y(t)=\frac{z^{-k}BR_i}{P_iB+Q_i\Delta A}w(t)+\frac{BE+Q_iC}{P_iB+Q_i\Delta A}\xi(t)\ . \tag{24}$$

For zero steady state errors, $P_i(z^{-1})$ and $R_i(z^{-1})$ can be chosen such that at z = 1, $P_i(1) = R_i(1)$, as $z^{-k}BR_i/(P_iB+Q_i\Delta A)\big|_{z=1}=1$. From (14), it is convenient to choose:

$$R_i=\frac{1+\lambda_{Pi}}{2}+\frac{1+\lambda_{Pi}}{2}z^{-1}\ . \tag{25}$$

(ii) The network weights $\hat{\theta}(0)$ are initialized to zero and the covariance matrix $P(0)$ to αI, where I is an identity matrix and α, a constant between 0.01 and 100.

(iii) The maximum range of the inputs can be set 20% above the normal operating range of the system, whilst the number of inner knots can be chosen to provide a smooth approximation of the nonlinearity of the system. Basis functions with second order, i.e. triangular basis functions, are generally adequate.

6 Control of Water Level Using *BSNN*-PI Controllers

A two-tank water level control system is shown in Fig. 2, where R_1 and R_2 are the valve resistances, and A_1 and A_2, the area of the tanks. Water is pumped into the first tank by a positive displacement pump driven by a D. C. motor and controlled by the voltage applied. The local linear model of the water level system is given by:

$$\frac{h_2(s)}{v_i(s)} = \frac{a_3}{a_1 s^2 + a_2 s + 1} = \frac{a_3}{(1+T_1 s)(1+T_2 s)} \cdot \qquad (26)$$

where $a_1 = R_1A_1R_2A_2$, $a_2 = R_1A_1 + R_2A_1 + R_2A_2$, $a_3 = K_{pump}R_2$, T_1 and T_2 are the time constants, h_2 and v_i are respectively small variations in the water level in the second tank and the applied voltage from a steady-state level. The following parameters are used in the simulation: K_{pump} = 36.98x10^{3}mm^3/sV, $A_1 = A_2$ =3167mm^2. Since the valve resistances, and hence the time constants are proportional to the square of the pressure difference across the valves, they are nonlinear, as given in Table 1. The simulation based on these values is therefore also nonlinear.

The *BSNN-PI* controller (18) is implemented with six triangular basis functions for $y(t)$, and two for $w(t)$. From (5), the number of weights in the network is 38. The range of the fuzzy sets for both $y(t)$ and $w(t)$ are between 0mm and 1000mm. The weights and the covariance matrix P are initialised to 0 and 0.1**I** respectively.

Since R_2 is proportional to the square root of the water level in the second tank, the gain of the system increases as the water level increases, but at a lower rate as the water level in tank 2 increases. It follows that λ_P and λ_Q need to be adjusted depending on the operating points. From (14), the controller gain becomes smaller, as λ_Q increases, or λ_P more negative. In contrast, larger controller gain is obtained, if λ_Q decreases, or λ_P less negative. Therefore, larger λ_Q and more negative λ_P should be chosen as the set-point is increased, as shown in Table 1. The performance of the *BSNN-PI* controller (18) is shown in fig.3, showing good closed-loop response despite the nonlinearity of the system.

Table 1. Parameters of two-tank system and choice of λ_P and λ_Q at speicifc operating points

H_2	R_1	R_2	T_1	T_2	λ_P	λ_Q
800	3.77	7.54	5.24	54.48	-0.91	10
600	3.27	6.53	4.54	47.18	-0.89	9
400	2.67	5.33	3.70	38.52	-0.83	7.7
200	1.89	3.77	2.62	27.24	-0.77	5.8

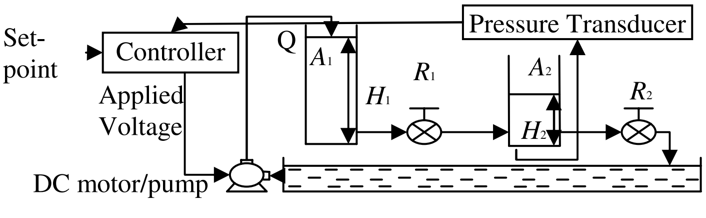


Fig. 2. Water level control system

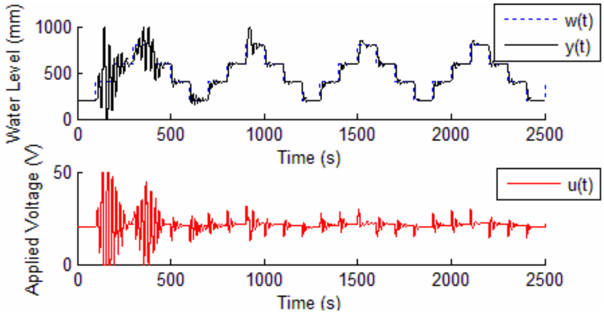


Fig. 3. Liquid level control using *BSNN*-PI controller under varying λ_P and λ_Q

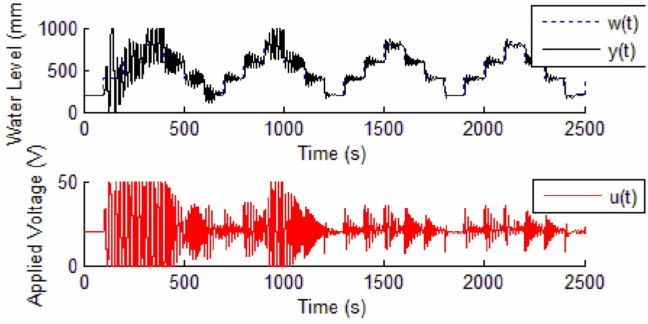


Fig. 4. Liquid level control using *BSNN*-PI controller under constant λ_P and λ_Q

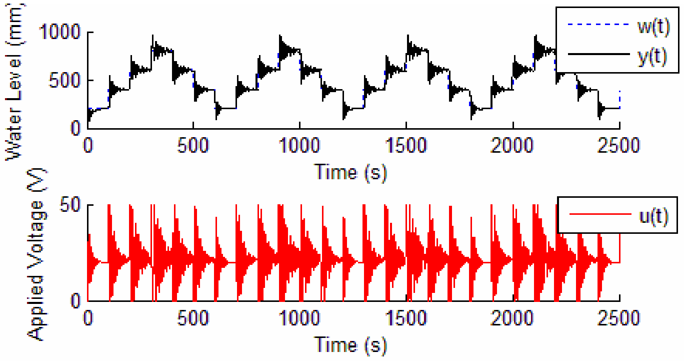


Fig. 5. Liquid level control using PI controller tuned by Ziegler-Nichols Tuning Rule based on Critical Gain and Critical Period

As a comparison, the simulation is repeated with *BSNN-PI* implemented using the same $P(z^{-1})$ and $Q(z^{-1})$ over the full operating rang with $P(z^{-1})$ = 1-0.83 z^{-1}, $Q(z^{-1})$ = 7.7 and $R(z^{-1})$ = $0.085+0.085z^{-1}$. The closed-loop response is shown in Fig. 4, showing larger oscillations at larger set-points as expected. The performance of the proposed *BSNN-PI* controller is also compared with PI controller tuned by Ziegler-Nichols

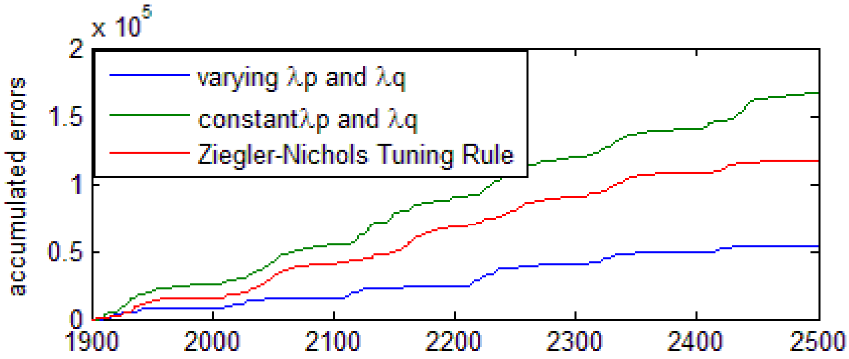


Fig. 6. Accumulated Errors from t=1900 to 2500

Tuning Rule based on Critical Gain and Critical Period. From Fig. 5, the closed-loop response is more oscillatory than that obtained by the *BSNN-PI* controller.

The accumulated sum of squared of the errors are computed compared for all three cases for the period from 1900 to 2500, as shown in Fig. 6. It is clear that the proposed *BSNN-PI* controller gives the smallest accumulated errors, and as expected, the worst result is obtained from the PI controller tuned by the Ziegler-Nichols Tuning Rule.

7 Conclusion

An adaptive nonlinear PI controller based on B-spline neurofuzzy network is proposed in this paper. The implementation of this controller is discussed, and guidelines for choosing the design parameters are presented. The proposed *BSNN-PI* controller is applied to control the nonlinear two-tank water level control system. It is shown that the *BSNN-PI* controller is able to control the processes after a short training period. The performance of the proposed *BSNN-PI* controller is compared with *BSNN-PI* with constant $P(z^{-1})$ and $Q(z^{-1})$, and PI controller tuned by the Ziegler-Nichols Tuning Rule based on Critical Gain and Critical Period, showing that the best result is obtained using the proposed controller. The proposed *BSNN-PI* controller can be extended to nonlinear PID controllers, which will be discussed in future works.

References

1. Ho, W.K., Hang, C.C., Zhou, J.H.: Performance and gain and phase margins of well-know PI tuning formulas. IEEE Transactions on Control Systems Technology 3(2), 245–248 (1995)
2. Gawthrop, P.J.: Self-tuning PID controllers: Algorithms and Implementation. IEEE Transaction on Automatic Control 31(3), 201–209 (1986)
3. Bittanti, S., Piroddi, L.: GMV technique for nonlinear control with neural networks. IEE Proceedings-D Control Theory and Applications 141(2), 57–69 (1994)
4. Zhang, T., Ge, S.S., Hang, C.C.: Neural-based direct adaptive control for a class of general nonlinear systems. International Journal of Systems Science 28(10), 1011–1020 (1997)

5. Narendra, K.S., Balakrishnan, J.: Adaptive control using multiple models. IEEE Transactions on Automatic Control 42(2), 171–187 (1997)
6. Chan, C.W., Liu, X.J., Yeung, W.K.: Neurofuzzy network based self-tuning control with offset eliminating. International Journal of Systems Science 34(2), 111–122 (2003)
7. Johansen, T.A., Foss, B.A.: Constructing NARMAX models using ARMAX models. International Journal of Control 58(5), 1125–1153 (1993)
8. Hong, X., Harris, C.J.: A neurofuzzy network knowledge extraction and extended Gram-Schmidt algorithm for model subspace decomposition. IEEE Transactions on Fuzzy Systems 11(4), 528–541 (2003)
9. Chan, C.W., Cheung, K.C., Yeung, W.K.: A computation-efficient on-line training algorithm for neurofuzzy networks. International Journal of Systems Science 31(3), 297–306 (2000)
10. Mok, H.T., Chan, C.W., Yeung, W.K.: Neurofuzzy network based adaptive nonlinear PI controllers. Control and Intelligent Systems 34(3), 216–224 (2006)

Appendix: Integrating Controller

By introducing an integrator in (2), the Auto-Regressive Integrated Moving Average model with eXogenous input (ARIMAX) model is obtained as follows,

$$A(z^{-1})y(t) = z^{-k}B(z^{-1})u(t) + C(z^{-1})\zeta(t)/\Delta. \tag{A1}$$

where $\Delta = (1-z^{-1})$ is the difference operator. The integrating control law is obtained by minimizing the square of a generalized system output:

$$\phi(t+k)= P(z^{-1})\, y(t+k) + Q(z^{-1})\Delta u(t) - R(z^{-1})w(t). \tag{A2}$$

where $P(z^{-1})$, $Q(z^{-1})$ and $R(z^{-1})$ are polynomials in z^{-1} with order n_p , n_q and n_r respectively. The integrating control law that minimizes (A2) is given below:

$$F(z^{-1})\,\Delta u(t) + G(z^{-1})y(t) + H(z^{-1})w(t)=0. \tag{A3}$$

where $P(z^{-1})C(z^{-1}) = E(z^{-1})\Delta A(z^{-1}) + z^{-k}G(z^{-1})$, $F(z^{-1}) = B(z^{-1})E(z^{-1}) + Q(z^{-1})C(z^{-1})$, and $H(z^{-1}) = -\,C(z^{-1})\,R(z^{-1})$.

Integrating Nonlinear Independent Component Analysis and Neural Network in Stock Price Prediction

Chi-Jie Lu[1],*, Chih-Chou Chiu[2], and Jung-Li Yang[2]

[1] Department of Industrial Engineering and Management, Ching Yun University,
Jung-Li, 320, Taoyuan Taiwan
[2] Institute of Commerce Automatic and Management,
National Taipei University of Technology, Taipei, 10608, Taipei, Taiwan
jerrylu@cyu.edu.tw

Abstract. In this research, a time series prediction model by integrating nonlinear independent component analysis (NLICA) and neural network is proposed for stock price. NLICA is a novel feature extraction technique to find independent sources given only observed data that are mixtures of the unknown sources, without prior knowledge of the mixing mechanisms. It assumes that the observed mixtures are the nonlinear combination of latent source signals. The proposed method first use NLICA as preprocessing to transform the input space composed of original time series data into the feature space consisting of independent components (ICs) representing underlying information/features of the original data. Then, the ICs are served as the input variables of the backpropagation neural network (BPN) to build prediction model. Experimental results on Nikkei 225 closing cash index show that the proposed model outperforms the integrated linear ICA and BPN model and the single BPN model without ICA preprocessing.

Keywords: Nonlinear independent component analysis, neural network, time series prediction, stock price.

1 Introduction

Stock price prediction is one of the most challenging application areas in organizations, firms, and private investors, whose activity is connected with financial markets. It is actually a difficult task because of the many correlated factors that become involved. These factors could be economic, political and even psychological. There is considerable interest in using computational intelligence techniques to analyze the extensive historical datasets on stock prices. A large number of successful applications have shown that neural networks can be a useful technique for stock price prediction due to their ability to capture subtle functional relationships among the empirical data even though the underlying relationships are unknown or hard to describe [1][2]. Unlike traditional statistical models, neural networks are data-driven and non-parametric models. They do not require strong model assumptions and can

* Corresponding author.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 614–623, 2009.

map any nonlinear function without a priori assumption about the properties of the data [3][4]. The most popular neural network training algorithm for stock price prediction is the backpropagation neural network (BPN) [1][3][4], which has a simple architecture but a powerful problem-solving ability. It motivates this study of using BPN for stock price prediction.

In the most existing prediction methods, the observed original values of prediction variables are usually directly used for building prediction models [1][2]. However, a stock price may be affected by some underlying factors like seasonal variations or economic events [5][6][7]. For example, in [5], the observed daily stock returns reflected the stock market reaction to a few factors such as release of various economic indicators, company news, psychological factors, government intervention, or political issues. Kiviluoto and Oja [6] showed that the underlying factors affected the cashflow time series of several stores belonging to the same retail chain were trends, holiday effects, seasonal variation and competition. Therefore, if the underlying/interesting information that can't be observed directly from the observed original data can be revealed through transforming the input space into the feature space with suitable feature extraction methods, the prediction performance will be improved by using the features to build prediction model.

Independent component analysis (ICA) is a novel feature extraction technique. It aims at recovering independent sources from their mixtures, without knowing the mixing procedure or any specific knowledge of the sources [8]. The independent sources, called independent components (ICs), are hidden information of the observable data. ICA model can be classified as linear ICA (LICA) and nonlinear ICA (NLICA) models. Linear ICA model is mainly based on the assumption that the observed mixture data are generated by a linear mixing from ICs. Conversely, the nonlinear ICA model assumes that the observations are a nonlinear combination of ICs. Linear ICA is suitable when it is reasonable to assume that the mixing mechanism is linear. It has been employed successfully in various fields of multivariate data processing, from signal processing and face recognition to time series prediction [5][6][7][8]. However, in many realistic cases, the observed mixture data (i.e. observed stock price data) might be a nonlinear mixture of latent source signals. Since the nonlinear mixing model is more realistic and practical, NLICA has the potential to become rather powerful tools. Nonlinear ICA remains an active field of research, few applications of NLICA are proposed in literatures [9][10][11].

In this study, a time series prediction model integrating NLICA and BPN is proposed for stock price. The proposed approach first uses NLICA on the input space composed of original forecasting variables into the feature space consisting of independent components representing underlying/hidden information of the original data. The hidden information of the original data could be discovered in these ICs. The ICs are then used as the input variables of the BPN for building the prediction model. In order to evaluate the performance of the proposed approach, the Nikkei 225 closing cash index is used as the illustrative example.

The rest of this paper is organized as follows. Section 2 gives brief overviews of nonlinear ICA and neural networks. The proposed model is described in Section 3. Section 4 presents the experimental results and this paper is concluded in Section 5.

2 Methodology

2.1 Nonlinear Independent Component Analysis

Linear ICA is becoming a well research area. In the linear ICA model [8], it is assumed that m measured variables, $\mathbf{x} = [x_1, x_2, \cdots, x_m]^T$ can be expressed as linear combinations of n unknown latent source components $\mathbf{s} = [s_1, \mathrm{s}_2, \cdots, \mathrm{s}_n]^T$:

$$\mathbf{x} = \sum_{j=1}^{n} \mathbf{a}_j s_j = \mathbf{As} \tag{1}$$

where $\mathbf{a}_j$ is the *j-th* row of unknown mixing matrix $\mathbf{A}$. Here, we assume $m \geq n$ for A to be full rank matrix. The vector $\mathbf{s}$ is the latent source data that cannot be directly observed from the observed mixture data **x**. The linear ICA aims to estimate the latent source components **s** and unknown mixing matrix **A** from **x** with appropriate assumptions on the statistical properties of the source distribution. Thus, linear ICA model intents to find a de-mixing matrix $\mathbf{W}$ such that

$$\mathbf{y} = \sum_{j=1}^{n} \mathbf{w}_j x_j = \mathbf{Wx}, \tag{2}$$

where $\mathbf{y} = [y_1, y_2, \cdots, y_n]^T$ is the independent component vector. The elements of **y** must be statistically independent, and are called independent components (ICs). The ICs are used to estimate the source components s_j. The vector $\mathbf{w}_j$ in equation (2) is the *j-th* row of the de-mixing matrix $\mathbf{W}$.

Since the nonlinear ICA model assumes that the observations are a nonlinear combination of latent sources, it can be formulated as the estimation of the following model [11]

$$\mathbf{x} = F(\mathbf{s}) \tag{3}$$

where **x** and **s** denote the data and source vector as before, and *F* is an unknown nonlinear transformation function. Assume now for simplicity that the number of independent components equals the number of mixtures. The nonlinear ICA problem then consists of finding a mapping $G : \Re^n \to \Re^n$ that yields components

$$\mathbf{y} = G(\mathbf{x}) \tag{4}$$

which are statistically independent.

Nonlinear ICA is still less studies than linear ICA, although several works already exist [12]. In this study, the MISEP method proposed by Almeida [11] is adapted to solve for the ICs. It is an extension of INFORMAX method, a well-known algorithm for performing linear ICA. MISEP extends the INFORMAX method in two ways: (1) it is able to perform nonlinear ICA, and (2) it uses adaptive nonlinearities at the output [11]. These nonlinearities are intimately related to the statistical distributions of the components, and the adaptively allows the method to deal with components with a wide range of distributions.

The MISEP tries to perform Eq.(4) according to a mutual information criterion. The mutual information of the components of **y** is defined as [11]

$$I(\mathbf{y}) = \sum H(y_i) - H(\mathbf{y}) \tag{5}$$

where H denotes Shannon's differential entropy, $H(\mathbf{y}) = -\int p(\mathbf{y}) \log p(\mathbf{y}) d\mathbf{y}$, for continuous variables; The mutual information $I(\mathbf{y})$ is non-negative, and is zero if the components of y are mutually statistically independent; $p(\mathbf{y})$ represents the probability density of the multidimensional random variable y; $p_i(y_i)$ will denote the marginal density of the *i-th* component of **y**.

The mutual information is not affected by invertible transformations made on a per-component basis. That is, if one performs invertible, possibly nonlinear, transformations on individual components of **y**, $z_i = \psi_i(y_i)$, then $I(\mathbf{z})=I(\mathbf{y})$. However, Eq.(5) shows that to estimate mutual information $I(\mathbf{y})$ one needs to estimate the marginal densities $p_i(y_i)$. In principle, the joint density $p(\mathbf{y})$ should also be estimated. In practical situations one normally has access only a finite set of observations **x**. From these, the MISEP method can compute a finite set of vectors $\mathbf{y} = G(\mathbf{x})$, given a transformation *G*, but it does not have access to the actual densities $p_i(y_i)$ [11]. These densities have to be estimated from those data. It estimates the components' cumulative probability functions jointly with the optimization of the transformation *G*. The resulting ICA method has the advantage of using a single, specialized multiplayer perceptron (MLP), optimized according to a single objective function, the output entropy. For more detail information about MISEP algorithm, please refer to Almeida [11].

2.2 Neural Networks

A neural network is a massively parallel system comprised of highly interconnected, interacting processing elements (often termed units, nodes or neurons) that are based on neurobiological models. Neural networks process information through the interactions of a large number of simple processing elements. Owing to its associated memory characteristic and its generalization capability, neural networks are found to be useful in modeling non-stationary processes [3][4]. A network's architecture is made up of the organization of nodes and the types of connections permitted. The network consists of a number of nodes connected by links. The nodes in the neural network can be divided into three layers: input, output and one or more hidden layers. The nodes in the input layer receive input signals from an external source and the nodes in the output layer provide the target output signals. Any layers between input and output layers are called hidden layers. Various network architectures and learning algorithms have been developed. The backpropagation neural network (BPN) is a feedforward network and is probably the most commonly used class of neural networks in financial time series forecasting and business [1][2].

BPN is essentially a gradient steepest descent training algorithm. For the gradient descent algorithm, the step size, called the learning rate, must be specified first. The learning rate is crucial for BPN since smaller learning rates tend to slow down the learning process before convergence while larger ones may cause network oscillation and the inability to converge. As to the issue of determining the appropriate network topology (the number of layers, the number of nodes in each layer, and the appropriate learning rates), please refer to McNelis [4], for more details. Detailed descriptions of using neural networks in the applications of forecasting and business can be found in McNelis [4], Vellido et al. [2] and Zhang et al. [1].

3 Proposed NLICA-BPN Prediction Model

This paper proposes a two-stage time series price prediction model by integrating NLICA and BPN (called NLICA-BPN model). The Schematic representation of the proposed prediction model is illustrated in Figure 1.

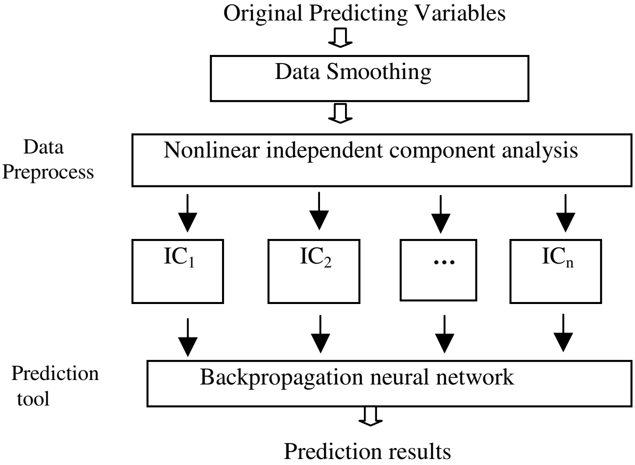

Original Predicting Variables
Data Smoothing
Data Preprocess
Nonlinear independent component analysis
IC$_1$
IC$_2$
...
IC$_n$
Prediction tool
Backpropagation neural network
Prediction results

Fig. 1. Schematic representation of the proposed prediction model

As shown in Figure 1, the first step of the proposed model is to smooth the forecasting variables since, in the conventional prediction model, the studied time series is usually smoothed in the first step through a suitable discrete differentiation preprocess. The original data $\mathbf{x}$ is processed by smoothing (logarithmic) step to have its smoothed form, $\mathbf{x}^s = \left(x_1^s, x_2^s, \ldots, x_m^s\right)^T$, using the following equations:

$$x_i^s = \log\left(x_i\right),\ i=1, 2, \ldots, m \tag{6}$$

The NLICA model is then used in the smoothed data $\mathbf{x}^s$ to estimate ICs, i.e. $\mathbf{s} = G(\mathbf{x}^s)$, where $\mathbf{s} = (s_1, s_2, \ldots, s_n)^T$ is the independent component vector consisting of n ICs. In the third step, the ICs contained hidden information of the prediction variables are used as input variables to construct BPN forecasting model. Finally, since the original data $\mathbf{x}$ is transformed into the smoothed form in the first step, the forecasted values should be inverse transformed to obtain the final predicted value for the original time series data $\mathbf{x}$.

In building the BPN forecasting model, since one hidden layer network is sufficient to model any complex system with desired accuracy [3], the designed BPN model in this study will have only one hidden layer. The performance of BPN is mainly affected by the setting of network topology, i.e. the number of nodes in each layer and learning rates. There are no general rules for the choice of network topology. The selection is usually based on the trial-and-error (or called cross-validation) method. In this study, the optimal network topology of the BPN model is determined by the trial-and error method.

4 Experimental Results

For evaluating the performance of the proposed NICA-BPN prediction model, the daily Nikkei 225 closing cash index is used in this study. In forecasting Nikkei 225 closing cash index, the Nikkei 225 index futures prices are used as forecasting variables since the futures price changes lead price changes of the cash market (Lee and Chen 2002, Lee and Chiu 2002). There are three Nikkei 255 index futures contracts traded on SGX-DT (Singapore Exchange-Derivative Trading Limited)($\mathbf{x}_1$), OSE (Osaka Securities Exchange)($\mathbf{x}_2$) and CME (Chicago Mercantile Exchange)($\mathbf{x}_3$) markets. The previous day's cash market closing index($\mathbf{x}_4$) is also an important variable for predicting the cash market opening price. Therefore, four forecasting variables are used for predicting the Nikkei 225 opening cash index. The daily data of futures and cash prices from February 2, 2004 to February 29, 2008 of the Nikkei 225 cash index provided by Bloomberg are collected in this study. There are totally 1004 data points in the dataset and the daily Nikkei 225 closing cash prices are shown in Figure 2. The daily values of the four forecasting variables are shown in Figure 3. The first 803 data points (79.98% of the total sample points) are used as the training sample while the remaining 201 data points (20.01% of the total sample points) are used as the testing sample.

The prediction results of the proposed model are compared to the BPN model without using NLICA preprocessing tool (called single BPN model), and integrated linear ICA and BPN model (called LICA-BPN model) which uses LICA as preprocessing tool. The prediction performance is evaluated using the following performance measures, namely, the root mean square error (RMSE), mean absolute difference (MAD), mean absolute percentage error (MAPE) and directional accuracy (DA).

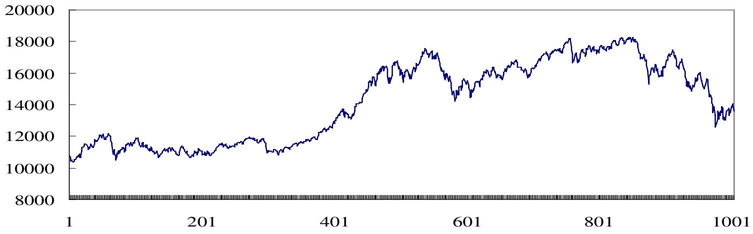


Fig. 2. The daily Nikkei 225 closing cash prices from 2/2/2004 to 2/29/2008

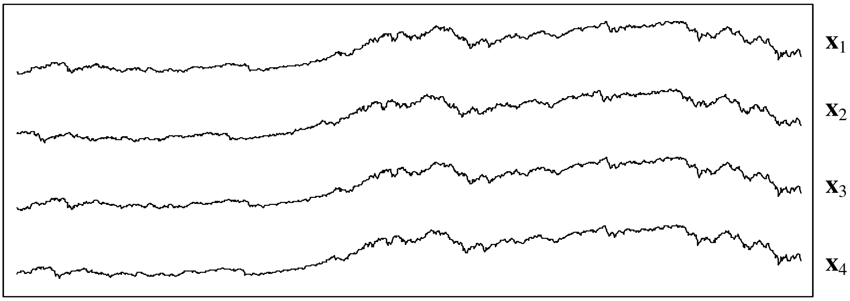


Fig. 3. The daily values of the four forecasting variables from 2/2/2004 to 2/29/2008

In the modeling of single BPN model, the four forecasting variables are directly used as input variables. The input layer has four nodes as four forecasting variables are used. Since there are no general rules for the choice of the number of the hidden layer, the number of hidden nodes to be tested was set to 7, 8, 9 and 10. And the network has only one output node, the forecasted closing cash price index. As lower learning rates tended to give the best network results [3], learning rates 0.01, 0.02 and 0.03 are tested during the training process.

For the proposed NLICA-BPN model, first, the original forecasting variables are smoothed and then passed to nonlinear ICA model to estimate ICs. Using the similar process, the LICA-BPN model uses the linear ICA model to estimate ICs from the smoothed forecasting variables. Figures 4 and 5, respectively, show the four ICs of the time series data in Figure 3 using NLICA and LICA models. It can be seen from Figures 4 and 5 that the ICs, regardless of using nonlinear ICA or linear ICA for the estimation, can be used to represent independent and different underlying factors like trends, seasonal variations, market news, government intervention, or economic events that affect the time series of exchange rates simultaneously. However, it also can be observed from the figures that the ICs of nonlinear ICA model can enhance/discover more detailed/hidden information than that of the ICs of linear ICA model. Then, the ICs in Figures 4 and 5 are, respectively, used for building BPN prediction models of the NLICA-BPN and LICA-BPN models. In building BPN models, like the single BPN model, the input layer has four nodes. The number of hidden nodes to be tested was 7, 8, 9 and 10. Again the network has only one output node, i.e. the forecasting closing cash price index, and learning rates 0.01, 0.02 and 0.03 are tested during the training process.

The convergence criteria used for training the NLICA-BPN, LICA-BPN and single BPN models are a root mean squared error (RMSE) less than or equal to 0.0001 or maximum of 1000 iterations. The network topology with the minimum testing RMSE is considered as the optimal network.

The testing results of the NLICA-BPN, LICA-BPN and single BPN models with combinations of different hidden nodes and learning rates are summarized in Tables 1, 2 and 3. From Table 1, it can be observed that the {4-7-1} topology with a learning rate of 0.03 gives the best forecasting result (minimum testing RMSE) and hence is the best topology setup for the proposed NLICA-BPN model in forecasting Nikkei 225 closing cash index. Here, {4-7-1} represents the four nodes in the input layer, 7 nodes in the hidden layer and one node in the output layer. Table 2 shows that the {4-8-1} topology with a learning rate of 0.03 gives the optimal topology setting for the LICA-BPN model. The {4-9-1} topology with a learning rate of 0.03 is the optimal topology setting for the single BPN model, as depicted in Table 3.

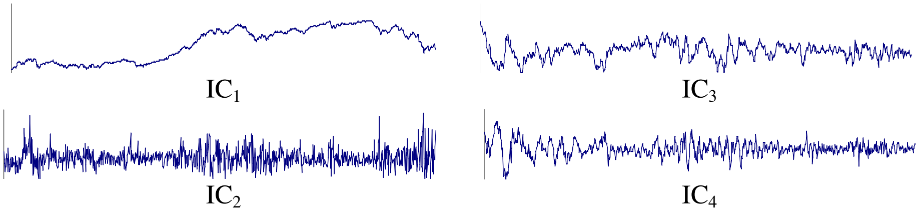


Fig. 4. Four ICs of the time series data in Fig.3 using NLICA

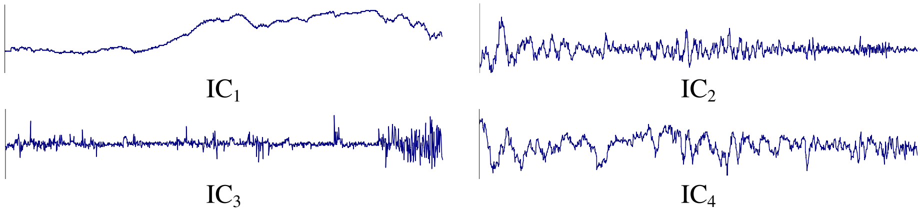


Fig. 5. The four ICs of the time series data in Fig.3 using LICA

Table 1. Model selection results of the NLICA-BPN model

Number of nodes in the hidden layer	Learning rate	Training RMSE	Testing RMSE
7	0.010	0.007743	0.014768
	0.020	0.007097	0.014356
	0.030	**0.002689**	**0.003055**
8	0.010	0.007160	0.013889
	0.020	0.006103	0.011577
	0.030	0.003681	0.006688
9	0.010	0.007814	0.014939
	0.020	0.006559	0.012979
	0.030	0.003523	0.005908
10	0.010	0.007691	0.014573
	0.020	0.007194	0.013623
	0.030	0.003776	0.007183

The Nikkei 225 closing cash price index forecast results using the proposed NLICA-BPN, LICA-BPN and single BPN models are computed and listed in Table 4. Table 4 depicts that the RMSE, MAD and MAPE of the proposed NLICA-BPN model are, respectively, 36.58, 27.87 and 0.19%. It can be observed that these values are smaller than those of the LICA-BPN and single BPN models. It indicates that there is a smaller deviation between the actual and predicted values using the proposed NLICA-BPN model. Moreover, compared to the LICA-BPN and the single BPN models, the NLICA-BPN model has the highest DA ratio which is 81.5%. DA provides a good measure of the consistency in prediction of the price direction. Thus, the proposed NLICA-SVR model provides a better forecasting result than the LICA-BPN and single BPN models in terms of prediction error and prediction accuracy.

Table 2. Model selection results of the LICA-BPN model

Number of nodes in the hidden layer	Learning rate	Training RMSE	Testing RMSE
7	0.01	0.012397	0.020699
	0.02	0.012223	0.020604
	0.03	0.009826	0.015797
8	0.010	0.012338	0.020442
	0.020	0.010988	0.018015
	0.030	**0.009817**	**0.015356**
9	0.010	0.012444	0.020723
	0.020	0.012279	0.020922
	0.030	0.009837	0.015548
10	0.010	0.012356	0.020614
	0.020	0.012180	0.020599
	0.030	0.009838	0.015558

Table 3. Model selection results of the single BPN model

Number of nodes in the hidden layer	Learning rate	Training RMSE	Testing RMSE
7	0.01	0.012062	0.021063
	0.02	0.010961	0.019226
	0.03	0.009785	0.017694
8	0.01	0.012140	0.021362
	0.02	0.011225	0.019748
	0.03	0.009987	0.018078
9	0.01	0.012022	0.020996
	0.02	0.011884	0.020519
	0.03	**0.009833**	**0.017355**
10	0.01	0.012075	0.021130
	0.02	0.011837	0.020574
	0.03	0.009876	0.017404

Table 4. The Nikkei 225 closing cash prices forecasting results using NLICA-BPN, LICA-BPN and single BPN models

	RMSE	MAD	MAPE	DS
NLICA-BPN model	**36.58**	**27.87**	**0.19%**	**81.5%**
LICA-BPN model	115.61	75.07	0.51%	73.2%
Single BPN model	128.78	92.87	0.65%	69.0%

5 Conclusion

This paper has presented a stock price prediction model by integrating nonlinear ICA and BPN. The Nikkei 225 closing cash index is used in this study for evaluating the performance of the proposed method. This study compared the proposed method with integrated linear ICA and BPN model, and single BPN model using prediction error and prediction accuracy as criteria. Experimental results showed that the proposed NLICA-BPN model can produce the lowest prediction error and the highest prediction accuracy in the datasets. It outperformed the integrated linear ICA and BPN model, and single BPN model. According to the experiments, it can be concluded that the nonlinear ICA can effective discover the hidden information of the original data and improve the prediction performance of BPN.

Acknowledgements

This research was partially supported by the National Science Council of the Republic of China under Grant Number NSC 97-2221-E-231-008.

References

1. Zhang, G., Patuwo, B.E., Hu, M.Y.: Forecasting with Artificial Neural Networks: the State of the Art. International Journal of Forecasting 14, 35–62 (1998)
2. Vellido, A., Lisboa, P.J.G., Vaughan, J.: Neural Networks in Business: a Survey of Applications (1992-1998). Expert Systems with Applications 17, 51–70 (1999)
3. Chauvin, Y., Rumelhart, D.E.: Backpropagation: Theory, Architectures, and Applications. Lawrence Erlbaum Associates, New Jersey (1995)
4. McNelis, P.D.: Neural Networks in Finance: Gaining Predictive Edge in the Market. Academic Press, New York (2004)
5. Mok, P.Y., Lam, K.P., Ng, H.S.: An ICA Design of Intraday Stock Prediction Models with Automatic Variable Selection. In: 2004 IEEE International Joint Conference on Neural Networks, pp. 2135–2140 (2004)
6. Kiviluoto, K., Oja, E.: Independent Component Analysis for Parallel Financial Time Series. In: 5th International Conference on Neural Information, Tokyo, Japan, pp. 895–898 (1998)
7. Back, A., Weigend, A.: Discovering Structure in Finance Using Independent Component Analysis. In: 5th International Conference on Neural Networks in Capital Market, pp. 15–17 (1997)
8. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley & Sons, New York (2001)
9. Haritopoulos, M., Yin, H., Allinson, N.M.: Image Denoising using Self-Organizing Map-Based Nonlinear Independent Component Analysis. Neural Network 15, 1985–1998 (2002)
10. Zhang, K., Chan, L.: Nonlinear Independent Component Analysis with Minimal Nonlinear Distort. In: 24th International Conference on Machine Learning, Oregon, pp. 1127–1134 (2007)
11. Almeida, L.B.: MISEP-Linear and Nonlinear ICA Based on Mutual Information. Journal of Machine Learning Reaching 4, 1297–1318 (2003)
12. Jutten, C., Karhunen, J.: Advances in Nonlinear Blind Source Separation. In: 4th Internal Symposium on Independent Component Analysis and Blind Signal Separation, Nara, Japan, pp. 245–256 (2003)

ELSA: A New Image Compression Using an Expanding-Leaf Segmentation Algorithm

Cheng-Fa Tsai and Jiun-Huang Ju

Department of Management Information Systems,
National Pingtung University of Science and Technology,
91201 Pingtung, Taiwan
{cftsai,m9656004}@mail.npust.edu.tw

Abstract. The development of multimedia has caused the heavy bandwidth load. Hence, digital content compression has become a significant topic presently. An appropriate codebook design method is a helpful and necessary principle for Vector Quantization (VQ). This work develops a new gray image compression algorithm named ELSA, witch exploits an expanding-leaf concept to determine the rough vectors (codebook) fast and utilizes the LBG for quality improvement in the end. Experimental results reveal that ELSA outperforms LBG, SOM and HSOM in terms of time-cost and image quality.

Keywords: Image compression, vector quantization, LBG, SOM, HSOM.

1 Introduction

As the result of the IT evolution, there are more and more network applications and demands, particularly for multimedia. Nevertheless, those digital contents may increase the need for storage capacity and network bandwidth. Consequently, data compression is a necessary and significant principle for those digital contents because of the smaller data capacity.

In general, Image compression schemes can be classified as lossy or lossless. Lossy compression models lower redundancy and irrelevancy by a series of transform in frequency domain to rebuild the image with the minimum errors. Redundancy reduction removes duplication signals, and irrelevancy reduction abates visually insignificant signals. Several lossy compressions, such as JPEG and JPEG2000, have been adopted. Lossless compressions, such as PNG and JPEG-LS, reconstruct the compressed images identically.

Spatial domain is another model that is dissimilar to frequency domain, and generally employs VQ (vector quantization) in a lossy compression. VQ is primarily utilized to minimize the number of codewords of a codebook to approach the original image. Namely, an image compression occurs by reserving a codebook composed of some codewords and an index-table that catalogs the index-values rather than raw codewords. Various VQ-based algorithms, such as LBG, SOM, HSOM and Tsai *et al.* [1], have been proposed to design the codebook appropriately and efficiently. Moreover, the proposed ELSA algorithm also belongs to this classification.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 624–633, 2009.

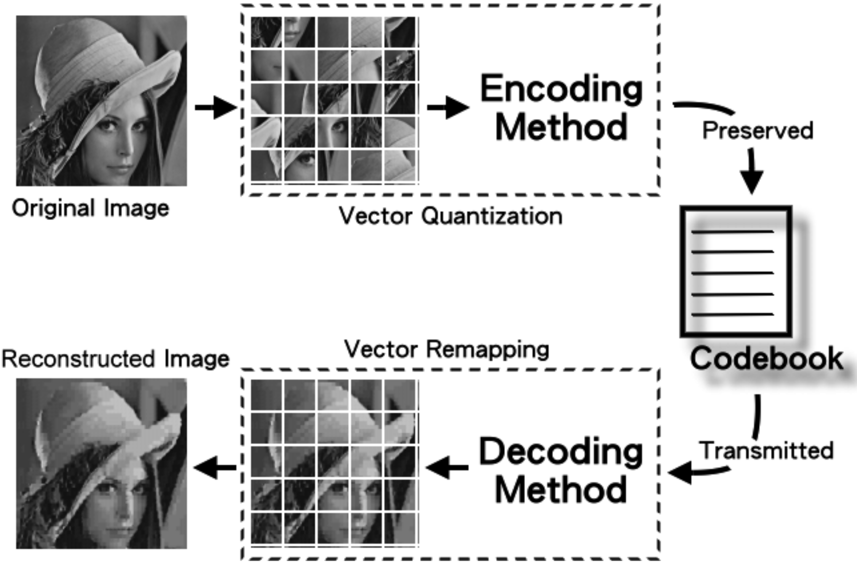


Fig. 1. The image compression and decompression procedure

LBG is fast and simple to implement, but has inconsistent coding, and is apt to becoming trapped in local optima. An artificial neural network structure called Self-Organizing Map (SOM) applied to image compression can achieve an excellent result, but it is very time-consuming. SOM trains the weights of several neurons, which it regards as codebook within specified training times. Since SOM gradually fine-tunes these neurons to derive global optima, it is named a full-search algorithm. Hierarchical Self-Organizing Map (HSOM) has been developed to decrease the time complexity of SOM. Although HSOM divides the entire compression process into a few stages to simplify the large scale and accelerating compression, but typically yields slightly worse quality solutions than SOM.

As shown in Fig. 1, an original image is first split into the smaller blocks according to parameter ***BlockSize***, which represents how many pixels a block requires. Accordingly, some encoding methods such as LBG, SOM or HSOM are performed with those blocks to generate the codebook that consists of ***n*** (given by parameter ***CodebookSize***) representative codewords. It is called compression that can conserve the codebook formed by encoding the image through Vector Quantization. On the other hand, to reconstruct an image from the codebook by Vector Remapping is namely denoted as decompression.

This investigation presents a new codebook design algorithm for gray image compression that adopts an **E**xpanding-**L**eaf **S**egmentation **A**lgorithm called **ELSA**. In the first stage, ELSA utilizes an arbitrary expanding leaf scheme and an effective estimation function to calculate the degree of distortion of each leaf, and thus may dynamically determine the rough representative vectors that form the initial codebook. In the final stage, a regular LBG is performed to train the initial codebook until a stopping criterion is met. The experimental results indicate that ELSA is with a quite better capacity than some existing well-known approaches.

The rest of this paper is organized as follows. Section 2 describes VQ, LBG, SOM, HSOM and objective measures. Section 3 then presents the proposed algorithm, called ELSA. Section 4 explains the experiments and analysis results. Conclusions are finally drawn in section 5, along with recommendations for future work.

2 Related Works

This section depicts in detail the concept of VQ and several VQ-based algorithms, such as LBG, SOM and HSOM, and discusses the properties of each of these algorithms. Lastly, the measure functions of the image quality are presented.

2.1 Vector Quantization

Vector Quantization (VQ), a representation developed by Gray and Gersho in 1980 [2]-[3] for approximating the original image via few vectors (codewords), was commonly employed in a lossy compression to preserve an image with a good quality and few acceptable distortions [4].

In Fig. 2(a), if two neighboring pixels have similar distribution, then these pixels are clustered into the same group, which has a representative vector $\boldsymbol{C}$ shown in Fig. 2(b). The representative vectors of each group then become the codewords for the codebook as listed in Fig. 2(c).

2.2 LBG

LBG, the best-known VQ technique for image compression, was developed by Linde, Buzo and Gray in 1980 [5] and was conceptually alike to the K-means data

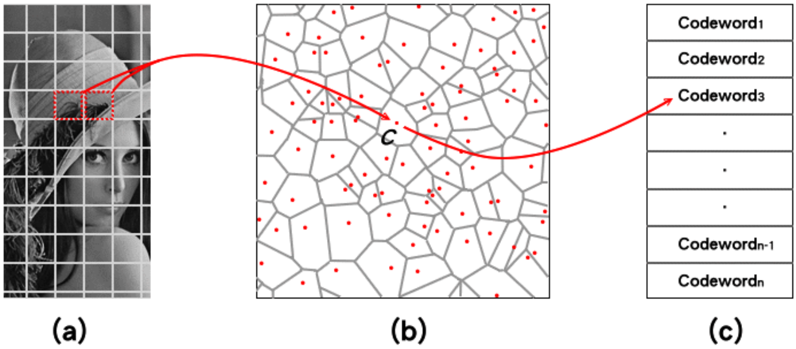


Fig. 2. The conception of Vector Quantization. (a) The blocked image (b) Codeword groups (c) The codebook constructed by $\boldsymbol{n}$ codewords.

clustering algorithm. LBG generates the codebook by continuously reclustering codewords until the average distortion is lower than a stipulated threshold (ϵ). The brief concept of LBG indicates that it can generate a codebook fast. However, the major restrictions of LBG are that it may be prone to get trapped at local optima, or yield an unstable result because of deciding the initial codebook randomly.

2.3 Self-Organizing Map

Self-Organizing Map (SOM), an unsupervised neural network, was developed by Kohonen in 1980 [6]-[7]. For applying SOM to image compression [8], it trains up the weighted values of neurons, which is to form the codebook. Furthermore, SOM fine-tunes these neurons very slowly and accurately to approach the global optima. As SOM undertakes a full-searching to get an excellent result, it is very time-consuming.

2.4 Hierarchical SOM

Even though SOM has a good quality of compressed image, its computational time is too high to design a codebook efficiently. Barbalho *et al.* first proposed a hierarchy-based SOM (HSOM) model for image compression [9]. In contrast to SOM, HSOM adopts a top-down scheme to accelerate the convergence.

Basically, HSOM has a lower computational time than SOM, but has two issues when applied to image compression. First, the number of neurons in each sub-group is fixed. Consequently, the number of neurons is unsatisfactory to express the codeword distribution appropriately. Second, each sub-group obtains its neurons from its superior neuron, probably causing those neurons to become trapped at local optima.

2.5 Measure Functions

The variation of a rebuilt image is easy to recognize for human vision. However, we still need some objective functions to measure the quality of a reconstructed image precisely. The Mean Square Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) are generally used to measure quality efficiently, and are defined by the following equations:

$$MSE = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} [i(x,y) - \bar{i}(x,y)]^2, \tag{1}$$

It is equivalent to

$$\sum_{g=1}^{Codewords} \frac{1}{GroupSize_g \times BlockSize} \sum_{n=1}^{GroupSize_g} \sum_{d=1}^{BlockSize} [i(x_{n_d}, y_{n_d}) - \bar{i}(x_{n_d}, y_{n_d})]^2 \tag{2}$$

$$PSNR = 10\log_{10}\frac{255^2}{MSE} \tag{3}$$

The Bits per Pixel (BPP), refers to the number of bit representing each pixel. The BPP is represented as Eqn. (4) below, where ***Codewords*** denotes the codebook size, and ***BlockSize*** indicates the dimension size of a codeword. For instance, the BPPs of 16 dimensional images consisting of 128 and 1024 codewords are 0.4375 and 0.625 each.

$$BPP = \frac{\log_2 Codewords}{BlockSize} \tag{4}$$

3 The Proposed ELSA Algorithm

As discussed earlier, although LBG is fast, it generates unstable clustering results because it randomly selects the initial codebooks. Moreover, although SOM and HSOM yield excellent result, they are much slower than other approaches. Accordingly, this work develops an image compression algorithm to combine the benefits of different algorithms.

This section describes the principle of the proposed image compression algorithm, namely ELSA. First, the ELSA algorithm settles the initial codebook by "codewords leaf expanding" where the codewords are dynamically produced by a modified LBG. "leaf growth estimation" is then employed to determine which leaf is qualified and need to expand. Finally, "codebook convergence" is adopted to stabilize the codewords and upgrade the quality of each leaf using a regular LBG. The detailed implementation of ELSA has three stages, which are described as follows:

(1) **Codewords leaf expanding:** In this stage, LBG is applied with the codeword that has maximum distortion currently to output two new clusters as Fig. 3 illustrated. In Fig. 3(a), the initial and single codeword '***A***' represents all vectors. In Fig. 3(b), as the previous codeword '***A***' is with maximum distortion, it is thus split into two new codewords '***A***' and '***B***'. In Fig. 3(c), all existing codewords (e.g. the previous codewords '***A***' and '***B***') whose distortion is the biggest must be divided into new codewords '***B***' and '***C***'. The incessant expanding is terminated until the desired number of codewords is obtained one by one.

(2) **Leaf growth estimation:** As we discussed the distortion in previous paragraph, ELSA determines which leaf (codeword) should expand based on a *Total Square Error (TSE)* measure as Eqn. (5):

$$TSE = \sum_{n=1}^{GroupSize}\ \sum_{d=1}^{BlockSize} [i(x_{n_d}, y_{n_d}) - \bar{i}(x_{n_d}, y_{n_d})]^2 \tag{5}$$

In Fig. 4, the MSE (noted in Eqn. (2)) of group ***T*** is given by $(L \times 3)/3/16 = L/16$ and that of group ***W*** is $(L \times 9)/9/16 = L/16$. Nevertheless, we consider

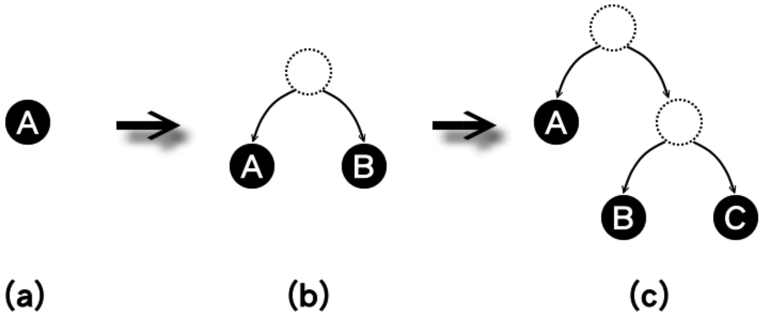


Fig. 3. The expanding leaf of codewords

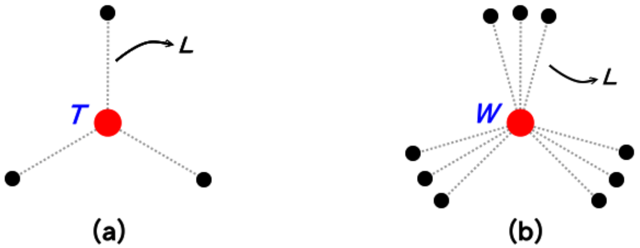


Fig. 4. Two 16-dimensional vector groups that have different distortion. (a) Vector group $\boldsymbol{T}$ consists of 4 vectors and has the same radius L (b) Vector group $\boldsymbol{W}$ consists of 8 vectors and has the same radius L.

the distortion of these two groups should be different, because they cannot both be expressed by only one represented vector (the centered vector in each group). Thus, this phase employs a TSE (observed in Eqn. (5)) function rather than the original MSE function. The TSE of group $\boldsymbol{T}$ is $(L \times 3) = 3L$, and that of group $\boldsymbol{W}$ is $(L \times 9) = 9L$. As a result, group $\boldsymbol{W}$ has a larger TSE than group $\boldsymbol{T}$, meaning that group $\boldsymbol{W}$ is allotted more codewords than group $\boldsymbol{T}$ so as to express vectors as appropriately as possible.

(3) **Codebook convergence:** After deriving the initial codebook through stage one, a regular LBG is utilized for fine-tuning the quality of the final codebook. In each iteration, ELSA calculates the recent MSE value of that codebook, and compares it with the previous MSE value. The terminal criterion of ELSA is that the variation of the recent and previous MSE value is below a tiny and specified threshold ($\boldsymbol{\epsilon}$). The equation of MSE variation is given as follows:

$$\varDelta D = |MSE(t) - MSE(t-1)|/MSE(t), \tag{6}$$

where $\boldsymbol{\varDelta D}$ means the variation of MSE; $|\cdot|$ represents the absolute value measure, and $\boldsymbol{t}$ denotes the iteration.

The pseudocode of ELSA algorithm is demonstrated below.

```
input  : $Pixels, CodebookSize, BlockSize, \epsilon$
output: $Codebook(codewords)$
$Blocks$= segment($Pixels$, $BlockSize$);
/*** Step1 Codewords leaf expanding ***/
while $codewords < CodebookSize$ do
    /*** Step2 Leaf growth estimation ***/
    find the most distorted codeword ($leaf0$) from $Blocks$ by Eqn. (5);
    recluster $leaf0$ into $leaf1$ and $leaf2$;
    remove $leaf0$ from $codewords$;
    add $leaf1$ and $leaf2$ to $codewords$;
end
/*** Step3 Codebook convergence ***/
while $variation > \epsilon$ do
    LBG( $codewords$ );
    calculate $variation$ by Eqn. (6);
end
output $codewords$;
```

Fig. 5. The pseudocode of ELSA algorithm

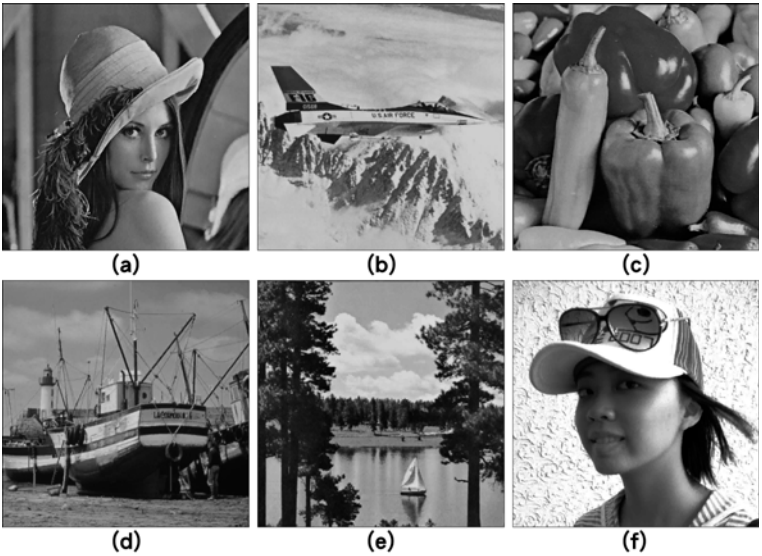
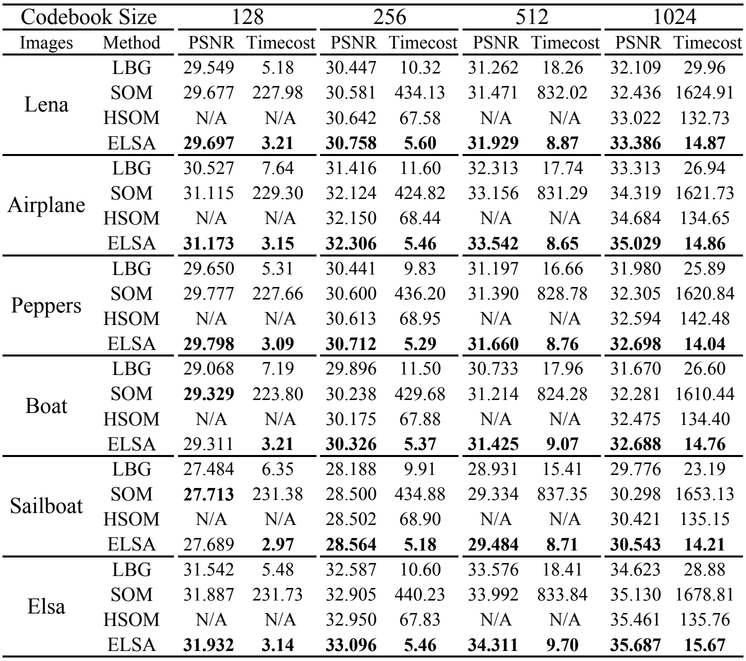


Fig. 6. The 1024-codewords and 16-dimension gray images reconstructed by ELSA. (a) Lena (b) Airplane (c) Peppers (d) Boat (e) Sailboat (f) Elsa.

4 Experimental Results

To verify the efficiency and quality of ELSA, a series of 30 independent runs was undertaken for five well-known and one private image, which are illustrated in Fig. 6. The algorithm was implemented in Java programming language using a personal computer with Intel Pentium4 3.2GHz CPU and 1GB RAM. Those images were tested at a size of 512×512 pixels and grayscale with 8 bits per pixel (BPP), i.e. 256 levels, and the block size was 4×4, namely 16 dimensions per vector. Significantly, HSOM cannot process an image with a codebook size that is not a square number, so its results were discarded in some tests, given as N/A (Not Available). The experimental results are listed below.

Fig. 7 depicts that ELSA has the best compression capability for images with large codebook sizes. Although ELSA performs a little worse than SOM in terms of compression quality in 128-codebook for images Boat and Sailboat, there are almost with the same compression quality and ELSA performs far fast than SOM. Notably, the most important thing is that the resulting quality in 128-codebook is too low to preserve the image. To help understand clearly the performance of all algorithms, the trend charts of PSNR and time-cost for the proposed ELSA and several existing approaches are illustrated in Fig. 8 and 9.



Codebook Size		128		256		512		1024	
Images	Method	PSNR	Timecost	PSNR	Timecost	PSNR	Timecost	PSNR	Timecost
Lena	LBG	29.549	5.18	30.447	10.32	31.262	18.26	32.109	29.96
	SOM	29.677	227.98	30.581	434.13	31.471	832.02	32.436	1624.91
	HSOM	N/A	N/A	30.642	67.58	N/A	N/A	33.022	132.73
	ELSA	**29.697**	**3.21**	**30.758**	**5.60**	**31.929**	**8.87**	**33.386**	**14.87**
Airplane	LBG	30.527	7.64	31.416	11.60	32.313	17.74	33.313	26.94
	SOM	31.115	229.30	32.124	424.82	33.156	831.29	34.319	1621.73
	HSOM	N/A	N/A	32.150	68.44	N/A	N/A	34.684	134.65
	ELSA	**31.173**	**3.15**	**32.306**	**5.46**	**33.542**	**8.65**	**35.029**	**14.86**
Peppers	LBG	29.650	5.31	30.441	9.83	31.197	16.66	31.980	25.89
	SOM	29.777	227.66	30.600	436.20	31.390	828.78	32.305	1620.84
	HSOM	N/A	N/A	30.613	68.95	N/A	N/A	32.594	142.48
	ELSA	**29.798**	**3.09**	**30.712**	**5.29**	**31.660**	**8.76**	**32.698**	**14.04**
Boat	LBG	29.068	7.19	29.896	11.50	30.733	17.96	31.670	26.60
	SOM	**29.329**	223.80	30.238	429.68	31.214	824.28	32.281	1610.44
	HSOM	N/A	N/A	30.175	67.88	N/A	N/A	32.475	134.40
	ELSA	29.311	**3.21**	**30.326**	**5.37**	**31.425**	**9.07**	**32.688**	**14.76**
Sailboat	LBG	27.484	6.35	28.188	9.91	28.931	15.41	29.776	23.19
	SOM	**27.713**	231.38	28.500	434.88	29.334	837.35	30.298	1653.13
	HSOM	N/A	N/A	28.502	68.90	N/A	N/A	30.421	135.15
	ELSA	27.689	**2.97**	**28.564**	**5.18**	**29.484**	**8.71**	**30.543**	**14.21**
Elsa	LBG	31.542	5.48	32.587	10.60	33.576	18.41	34.623	28.88
	SOM	31.887	231.73	32.905	440.23	33.992	833.84	35.130	1678.81
	HSOM	N/A	N/A	32.950	67.83	N/A	N/A	35.461	135.76
	ELSA	**31.932**	**3.14**	**33.096**	**5.46**	**34.311**	**9.70**	**35.687**	**15.67**

Fig. 7. Comparison of PSNR (in dB, higher is better) and time-cost (in second, lower is better) of reconstructed images for the proposed ELSA and other algorithms. The bold types denote the best results.

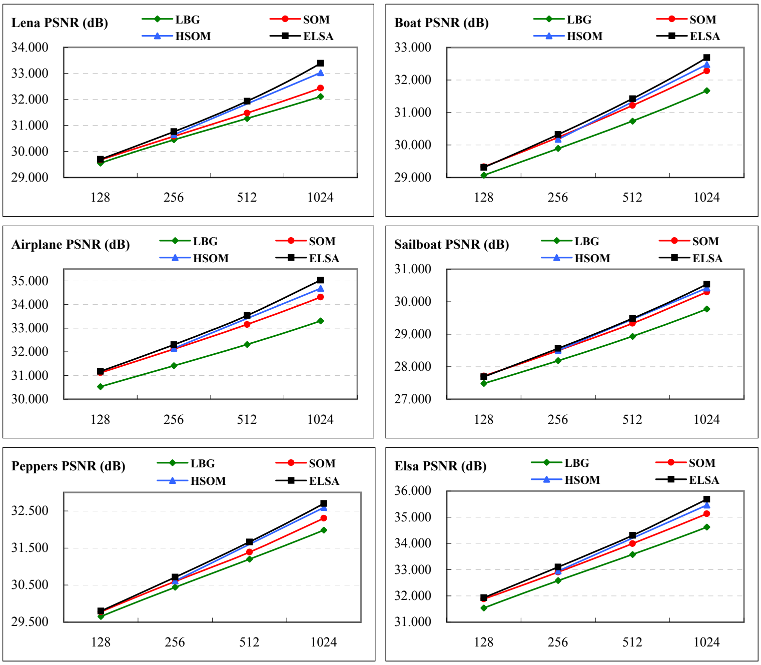


Fig. 8. The trend chart of PSNR for the proposed ELSA and other approaches

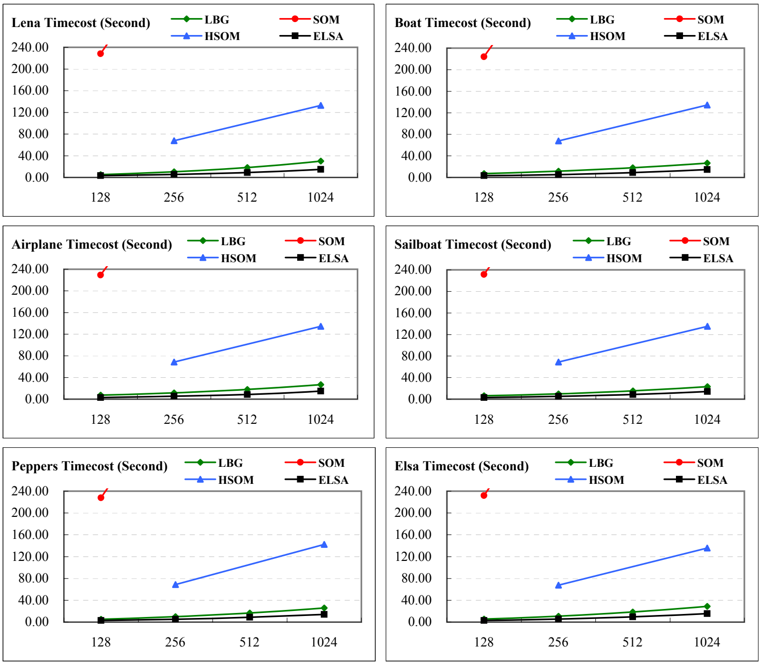


Fig. 9. The trend chart of time-cost for the proposed ELSA and other approaches

5 Conclusions

This work presents a superior and efficient codebook design technique for grayscale image compression. The proposed algorithm, named ELSA, first employs an expanding leaf principle to determine the initial codebook, and then adopts a regular LBG to stabilize and reinforce the initial codebook. In contrast to SOM and HSOM, the number of parameters for the proposed ELSA algorithm is as less as LBG (all they need is ϵ). Notably, although a smaller codebook size yields a higher compression ratio, it is not worth considering since the resulting quality is too low to preserve the image. Experimental results demonstrate that the proposed ELSA performs quite better and has a far shorter computational time than LBG, SOM and HSOM.

Acknowledgment. The author would like to thank the National Science Council of Republic of China, Taiwan for financially supporting this research under contract no. NSC 96-2221-E-020-027.

References

1. Tsai, C.F., Jhuang, C.A., Liu, C.W.: Gray Image Compression Using New Hierarchical Self-Organizing Map Technique. In: Proceedings of the 3rd International Conference on Innovative Computing Information and Control, pp. 544–549 (2008)
2. Gray, R.M.: Vector Quantization. IEEE ASSP Magazine 1(2), 4–29 (1984)
3. Gersho, A., Gray, R.M.: Vector Quantization and Signal Compression. Kluwer Academic Publishers, Boston (1991)
4. Sayood, K.: Introduction to Data Compression, 2nd edn. Morgan Kaufmann Publishers, San Francisco (2000)
5. Linde, Y., Buzo, A., Gray, R.: An Algorithm for Vector Quantizer Design. IEEE Trans. on Communications 28, 84–95 (1980)
6. Kohonen, T.: Self-Organizing Maps. Springer, Heidelberg (1995)
7. Kohonen, T.: The Self-Organizing Map. Proceedings of the IEEE 78(9), 1464–1480 (1990)
8. Madeiro, F., Vilar, R., Neto, B.: A Self-organizing Algorithm for Image Compression. IEEE Trans. on Neural Networks 28, 146–150 (1998)
9. Barbalho, M., Duarte, A., Neto, D., Costa, F., Netto, A.: Hierarchical SOM Applied to Image Compression. In: Proceedings of International Joint Conference on Neural Networks, vol. 1, pp. 442–447 (2001)

A Dynamic Rearrangement Mechanism of Web Page Layouts Using Web Agents

Masato Nakamura, Shohei Asami, Tadachika Ozono, and Toramatsu Shintani

Dept. of Computer Science and Engineering,
Graduate School of Engineering, Nagoya Institute of Technology,
Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555 Japan
{masato,asami,ozono,tora}@toralab.ics.nitech.ac.jp

Abstract. This paper describes a dynamic rearrangement mechanism of Web page layouts. The aim of this research is to provide webmasters or administrators with a means of determining easily the best suited content layout of their Web pages in a short period of time by automatically testing and relocating the contents with a trend of user actions. We will become able to modify dynamically the structure and layout of Web pages on the client browsers using the information of user behaviors. Our method provides the means of changing the contents inside or between Web pages and updating the contents on Web browsers, and it shorten the period of time for optimization.

1 Introduction

We are developing a system which will assist in improving content layouts of Web pages. Improving a Web page is generally done using iterative development with simple heuristics[6], the issue being that this kind of technique requires so much time that it is inefficient. One ideal way to improve a Web page is to provide the visitors with a Web page of which the contents are dynamically laid out on the fly. However, it is difficult to automate the work, because Web page design has an artistic side to it.

The aim of this research is to provide webmasters or administrators with a means of determining easily the best suited content layout of their Web pages by automatically testing and relocating the contents with a trend of user actions. In the paper, we call it Web page dynamic rearrangement.

We will become able to modify dynamically the structure and layout of Web pages on the client browsers. To this end, we will be using a Web agent running as an autonomous software inside the visitors' Web browser. Not only those webmasters will become able to dynamically rearrange their Web page layouts on generated Web pages, such as Content Management System (CMS) pages, but also on simple static Web pages. The current trend in research, regarding systems which would assist webmasters in rearranging their contents layout, is either focused on using statistics of which contents tend to be clicked more than others[5], or capitalizing on well-known sequential patterns[7]. Both of these

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 634–643, 2009.

methods will mostly require only a server-side implementation, whereas our research also focuses on using client-side agents. The reason why we need Web agents, as detailed later, is that we want to push further this usage of statistics and patterns, by generating far more detailed reports of the user behaviors; the best place to run for an agent with this aim is the visitor's physical point of view (i.e the client-side). On any Web server, it is possible to get detailed user access statistics, including page views and the like. However, those numbers take account of accesses from crawlers or despiteous users.

When rebuilding the structure of a Web page on the fly, it is important to take account of multiple kinds of terminals. As a system meant to be able to dynamically rearrange Web page layouts on any likely kind of browser, we prototyped two systems: one for a PC browser and one for a browser in a cellular phone. In the first case, we use push-type content delivery mechanism, because we attach a great importance to real-time information delivery. In the latter case, we didn't use the mechanism, because there are two important needs when considering a cellular phone environment: making the number of accesses to the server as low as possible, and the difficulty of browsing on a cellular phone.

In the next section, we describe how the Web agents operate when performing Web evaluation. In Section 3, we propose the method for rearranging Web page layouts in which we describe the mechanism of the push-type content delivery with Web agents, the details about Web evaluation, and a method to implement the dynamic rearrangement mechanism. In Section 4, we show a practical application how effectively the mechanism can be used in the emergency information delivery system.

2 Accurate Web Evaluation on a Client

Means of efficiently showing contents basically depend on those contents' layout, which indulge users into behaving as expected by the Web site's company or developers. Commonly expected behaviors include scenarios such as purchasing goods, petition of a catalog, and so on. To put it another way, if a user buys something using a Web application or Web site where a "purchase" link is shown, one can say that the content is exposed efficiently.

There are numerous characteristics used in Web evaluation, such as page views, click rate, number of unique users/IPs. What is described as page view, for example, is every single access to Web content through a browser request. Click rate, on the other side, represents the amount of user clicks to access various contents. Finally, the number of unique users browsing a Web site is obtained by substracting the number of accesses from the same user to the total number or page views. Basically, those variables cannot tell explicitly whether contents were effectively displayed to the user or not, since they simply count the number of requests to the server. In other words, it is difficult to accurately log "real" page views and user behavior, and other types of accesses or impressions, such as requests made through a Web proxy, pages viewed using a browser cache, and contents accessed by non-human clients such as automatons or bots. Commonly occurring issues include prematurate termination of communication between server and clients: the request has been made and logged, but what actually happened on the client side (has the page being shown to the visitor?) remains unknown and impossible to evaluate from the server side. Therefore, accurate user behavior evaluation on the server side remains difficult.

As stated before, server-side logging of accesses to Web contents has another defect known as Web access attacks: indeed, dummy accesses to servers by programs may cause the result estimated by the access logs to be far more greater than the actual number of page impressions, since accesses from a programs such as Web crawlers are always logged.

One way to avoid logging accesses from such crawlers, such as spam bots, would be not to log intensive requests: using the HTTP header information, it becomes possible to notice requests coming too rapidly from a same IP address, which is not likely human behavior. But there is a flaw in this kind of strategy: accesses made from a proxy that internet service providers offer may be looked upon as such "intensive requests" by the server. This one of the main reasons why this kind of server-side intelligent logging and user behavior evaluation remains very difficult to implement in practical applications. Additionally, HTTP headers are easily spoofed by malicious automaton programs, and a proxy server usually rewrites HTTP headers before routing requests. Moreover, those issues become more and more troublesome when the number of accesses from computers all across the globe is likely to increase.

We develop a push-type advertisement delivery system called the Wisdom Ad Balloon[2]. This is why the Wisdom Ad Balloon features a functionality to evaluate real page accesses on the client side, which we have developed to solve this problem; it is implemented using a push-type content delivery system[4]. To sum up, Web servers will gain the ability to log accesses and evaluate user behaviors only when those contents are actually being displayed on Web browsers and browsed by humans, not software agents.

Firstly, we accurately evaluate the browsing usability of displayed contents in Web browsers. That is to say, we measure the state of events such as: page scrolling, mouse movement, and so on. Secondly, we analyze, and therefore become able to modelize a user's behavior from a great deal of event data. Research projects nowadays show that machine leaning makes it likely possible to implement such functionalities with ease[8].

3 Rearranging Web Page Layouts

3.1 The Push-Type Content Delivery Mechanism

In this study, we use MiSpider[1], which is an agent system framework running on Web browsers. The MiSpider agent provides the Web developers with the ability of using intelligent Web services on Web browsers. Indeed, problems occur in achieving intelligent Web services on Web browsers. First, since the Web agent is nested on each page, it is difficult to provide continuous services: the agent will disappear and be replaced by a new one as the old page is itself replaced by a new one when browsing. Second, the resources on Web browsers are limited, and

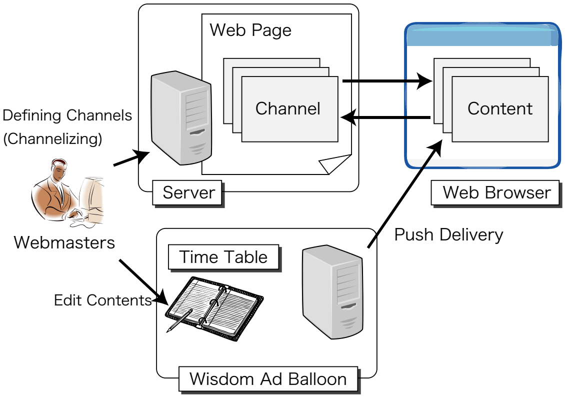


Fig. 1. Push-type Content Delivery Mechanism on Wisdom Ad Balloon

Web browsers need to cooperate with a Web server. However, it is difficult to communicate with a Web server at will. These problems are both solved by the MiSpider agent framework. Any agent built with this framework is persistent: an agent's data and/or states are not reinitialized when users browse through pages. Moreover, agents have a message passing capability to communicate between one another. Using these features, our system will enable real time content delivery on a Web browser.

We develop a push-type advertisement delivery system called the Wisdom Ad Balloon. Figure 1 shows the outline of the push-type content delivery mechanism on the Wisdom Ad Balloon. The system uses MiSpider agents and can use a push-type content delivery mechanism. The push-type delivery means the mechanism which allows to replace contents on a Web page without reloading the page. In order to use this agent system, webmasters simply have to write the HTML tag for contents delivery in their Web pages. We call the tag the "channel". The term "channel" is used to describe an area where contents delivered by the push-type delivery mechanism are displayed. Agents provide three features: collecting user behaviors, managing the contents delivery status, and reflecting the contents newly added or renewed to a Web page. Those webmasters can manage the push-type content delivery mechanism by using a time table for content management.

We make use of this push-type content delivery system; we also developed a system for extracting contents from a Web page and add it to content delivery system. This system features a function for sharing the contents among Web pages easily, to be used by webmasters and Web site administrators. Moreover, they become able to set up the display interval and parameters of every single push-type content.

3.2 The Dynamic Rearranging Mechanism

Since a content layout is likely to influence the user's behavior, many researchers and Web engineers make a great deal of effort in optimizing this layout. For

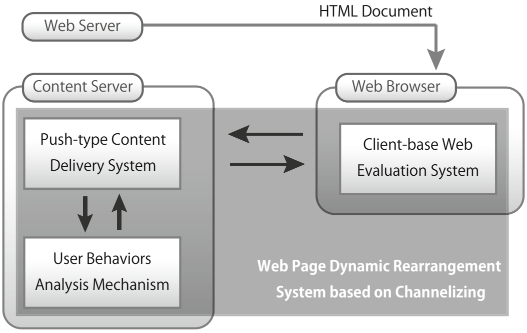

Web Server
HTML Document
Content Server
Web Browser
Push-type Content Delivery System
Client-base Web Evaluation System
User Behaviors Analysis Mechanism
Web Page Dynamic Rearrangement System based on Channelizing

Fig. 2. The outline of the dynamic rearranging system

instance, the left-top area of Web pages typically tends to attract a reader's attention more than the right-bottom area, since visitors usually read from left to right and from top to bottom. Furthermore, when a Web page is opened in a browser, this right-bottom part is likely to be hidden unless the reader scrolls down the page. Although such simple heuristics are well-known, the relation between this layout and the user's behavior is more difficult to figure out. If we were, for example, to simply switch the contents displayed on either side of a typical Web page, the user's behavior would likely change.

According to those facts, in many cases when rearranging content layouts on Web pages, iterative methods are used, which are inefficient in the long term. Therefore, we aim to use behavior evaluation to automate this process. Among practical applications, there is an implementation model known as split run test, which is a technique for optimizing the page's layout by switching contents on it. To put it simply, switching two different contents displayed in different places on the page will easily let us know which one is best, depending on what we want the user to do. The drawback of this technique is that it may take a great amount of time to evaluate the resulting user behavior, because statistics are likely to be used with great amount of data over long time spans.

This study features a short-term and accurate Web page optimization by evaluating user behaviors while they are actually browsing. As a sample implementation, let us introduce an emergency information delivery service, which needs accurate information on the user's behavior, in order to display the emergency information in the best place available while the user is browsing. If such a system was to feature advertisement instead of emergency broadcasts, short-term results would also be necessary for the same reason, in order to raise awareness. It is best suited for swiftly making advances in our research.

Let us explain more about this client-side evaluation system, and about the Web page's featured dynamic rearrangement mechanism, based on *channelizing* and implemented in the above-stated practical application. The "channelizing" refers to moving contents from another area of a Web page to a channel. We will describe the details about the channel at next subsection.

Figure 2 shows the outline of the system realizing the dynamic rearranging mechanism. The system consists of the push-type content delivery system, the client-base Web evaluation system and a Web page's dynamic rearrangement system based on channelizing. The first system runs on a Web server, managing Web agents and contents which are then delivered to the client. The second system runs in a Web browser and logs the user's actions and Web page history. The data that all agents collect gets reported to the first system, and then sent to the user behaviors data analysis mechanism which determines the most suited content layout.

3.3 The Channelizing Mechanism

The systems which generates dynamic layouts include CMS as represented by blog systems. For most users, CMS is a very useful tool, as demonstrated by the current trend about blog management systems. Such CMS blog management systems automatically generate Web pages with typical layout templates. If the blog's administrator or webmaster wants to further customize his layout, and add features such as dynamic Web page rearrangement, it is likely that he will have to customize his CMS program from the inside. This is, as expected, very difficult for the majority of people using the Web.

As stated before, we will use the channelizing mechanism to solve this problem and implement dynamic layout rearrangement. The term "dynamic" is not only used to qualify the rearrange-and-move feature and the visual appeal that comes with it, but also the push-type content delivery system. Indeed, we empower users with the ability to switch contents at any time. If the user is browsing not only one page, but multiple pages using our push-type delivery system, opened in different windows or tabs of the Web browser at the same time, channelizing will take effect not only on the channels of one of those pages, but on all the channels of the currently opened pages.

We describe a system called the Web clip system for channelizing. Figure 3 shows the process of channelizing on a Web page. The Web clip system extracts

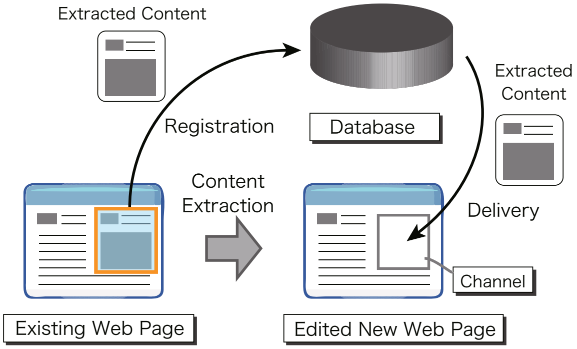


Fig. 3. Channelizing on a Web page

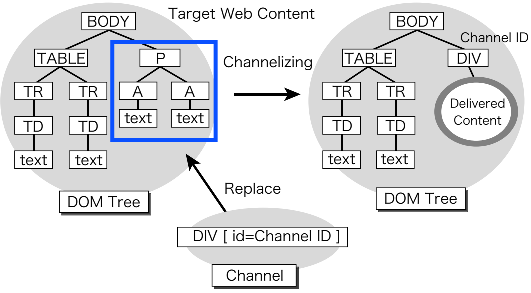
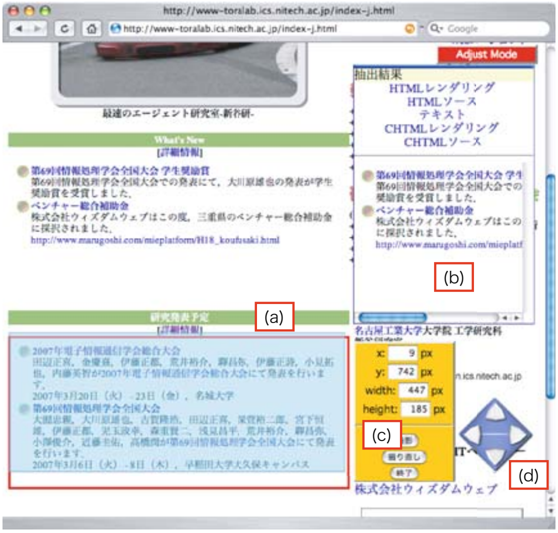


Fig. 4. Replacement of the DOM node



最速のエージェント研究室-新谷研-

What's New

[詳細情報]

第69回情報処理学会全国大会 学生奨励賞
第69回情報処理学会全国大会での発表にて，大川原康也の発表が学生奨励賞を受賞しました．

ベンチャー総合補助金
株式会社ウィズダムウェブはこの度，三重県のベンチャー総合補助金に採択されました．
http://www.marugoshi.com/mieplatform/H18_koufusaki.html

研究発表予定

[詳細情報]

2007年電子情報通信学会総合大会
田辺正寛，金慶燕，伊藤正都，荒井裕介，[illegible]，伊藤正詩，小見拓也，内藤英智が2007年電子情報通信学会総合大会にて発表を行います．
2007年3月20日（火）- 23日（金），名城大学

第69回情報処理学会全国大会
大囲忠親，大川原康也，古賀隆浩，田辺正寛，[illegible]，宮下恒雄，伊藤正都，[illegible]，森重賢二，浅見昌平，荒井裕介，[illegible]，小澤俊介，近藤圭佑，高橋周が第69回情報処理学会全国大会にて発表を行います．
2007年3月6日（火）- 8日（木），早稲田大学大久保キャンパス

第69回情報処理学会全国大会 学生
第69回情報処理学会全国大会での
奨励賞を受賞しました．
ベンチャー総合補助金
株式会社ウィズダムウェブはこの
に採択されました．
http://www.marugoshi.com/mieplatf

名古屋工業大学大学院 工学研究科

n.ics.nitech.ac.jp

株式会社ウィズダムウェブ

Fig. 5. Example of Channelizing in the Web clip system

the contents of the user's selection's closest DOM node (as described later), and then exchange this DOM node for an empty DIV tag which holds a channel ID in its own ID attribute. The channel ID's purpose is to store a string which will be used to identify the area where the DIV tag is located on the page. The extracted contents (when the node is replaced) get saved into a database on the content management server, as does the channel ID, in order to be able to identify the area where we will push again those saved contents dynamically. In other words, the contents saved upon the process are delivered with the push-type content delivery mechanism, and the Web page is displayed as before, but it now has the ability to dynamically change its content in the channel.

Figure 4 shows an example of what occurs when replacing an extracted DOM node within a channel. The P child node of the BODY node is the top of target nodes for channelizing. The right part of the figure shows the state of the system when the P node gets replaced with a newly born channel's DIV node. The contents of the P node are saved to the database, and then get delivered back to the channel node.

Figure 5 shows an execution example of channelizing using the Web clip system. The mechanism is used as follows: First, the webmaster or administrator decides which area he wants to transform into a channel, such as the (a) in the figure. The (b) holds the results of the extraction from the DOM tree. The (c) lets the webmaster edit the properties of the area he has just made into a channel and save the DOM data to the database on the server. The (d) button is the interface for resizing the channel area.

4 Practical Application: Wisdom Alert

In this section, we describe an emergency information delivery system called the Wisdom Alert. The Wisdom Alert uses the push-type content delivery mechanism along with the Web agents used for user behavior evaluation. The system enables a user to get real-time dynamic information delivery to the channels located in Web pages.

Figure 6 shows the outline of the system. Wisdom Ad Balloon is used as a core system for push-type content delivery, along with an information compilation system based on a card model for cellular phones: the Wisdom Card[3]. Wisdom Ad Balloon provides us with the features of real-time push-type content delivery and the Web agents that act as user behavior evaluators; it also empowers us with the ability to manage the contents displayed in our channels. When it comes to cellular phones, the contents are compiled by the Wisdom Card system and then delivered. The details of the Wisdom Card is omitted because of space limitations.

The Web agents control the contents in the channels and contents delivery according to the assignments set up in the content management server. To be more precise, the Web agents change the contents in the channels, following the server's strategy and instructions. Messages coming from the server hold details about when and where new contents should be displayed. The Web agents also log the user's behavior while browsing Web pages, and report them to the server. The application running on the server side then analyses the data and modelizes the user's behavior in order to elaborate its strategy.

A major asset to the system is that it is possible to use the push-type content delivery mechanism on a static Web page, even by webmasters or administrators who don't have any technical knowledge about Web development. The process of channelizing and push-type content delivery is as follows. First, webmasters use the Web clip system for channelizing as Figure 5. Second, they select a content from the view of contents using the delivery controller to deliver as shown on Figure 7. The delivery controller is the visceral interface for the push-type content delivery. Third, they push the delivery button of the controller to deliver the content.

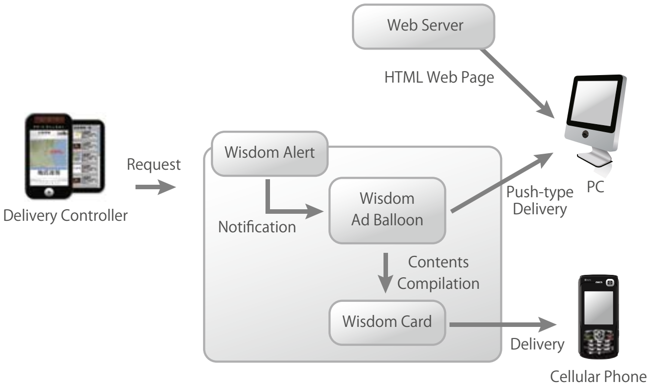


Fig. 6. The outline of the Wisdom Alert

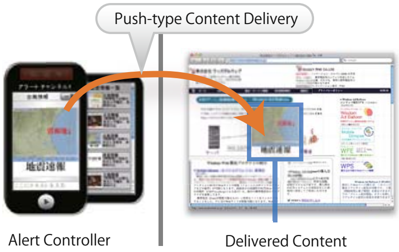


Fig. 7. The Alert Controller

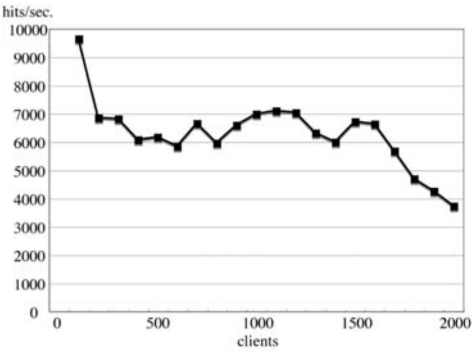


Fig. 8. The system performance of the Wisdom Alert

In the Wisdom Alert, the high performance is required to realize real-time content delivery. Figure 8 shows how many accesses the system is capable of dealing with when many clients concurrently connect to it. The following hardware was used in the test; dual 2.8GHz Quad-Core CPUs with 2GB RAM. As shown in Figure 8, the system is able to process many accesses in a short period of time.

5 Conclusions

In this paper, we propose a dynamic rearrangement mechanism for realizing Web page layouts using the Web agents. The Web agents are used to dynamically rearrange Web page layouts, but also collect user behaviors on a Web page. As for Web page dynamic rearranging and channelizing, we explained our strategy for implementing the DOM extraction system called Web clip. Additionally, we have stated our means of action by showing a sample implementation, the Wisdom Alert real-time emergency information system. It supports two kinds of terminals; a PC browser and a browser in a cellular phone. The contents delivered to cellular phones are optimized by an information compilation system called the Wisdom Card.

From now on, there is a forthcoming challenge that we will take on: it is to create a system able to analyze user behaviors for Web page optimization.

References

1. Fukagaya, Y., Ozono, T., Ito, T., Shintani, T.: Mispider: a continuous agent on web pages. In: WWW 2005: Special interest tracks and posters of the 14th international conference on World Wide Web, pp. 1008–1009. ACM, New York (2005)
2. Mukai, Y., Ozono, T., Ito, T., Shintani, T.: Dynamic page construction system based on push type transmission of information. In: Proceedings of the 70th National Convention of IPSJ (2005)
3. Ozono, T., Kakimoto, H., Sano, H., Hirata, N., Shintani, T.: An information compilation system based on card model for mobile phones. In: Proceedings of the 7th Forum on Information Technology 2008 (FIT 2008) (2008)
4. Ozono, T., Shintani, T.: Implementation of measures of effectiveness for web contents against spams. In: Proceedings of the 22nd Annual Conference of the Japanese Society for Artificial Intelligence (JSAI) (2008)
5. Richardson, M., Dominowska, E., Ragno, R.: Predicting clicks: estimating the click-through rate for new ads. In: WWW 2007: Proceedings of the 16th international conference on World Wide Web, pp. 521–530. ACM, New York (2007)
6. Shneiderman, B.: Designing the User Interface: Strategies for effective Human-Computer-Interaction. Addison-Wesley, Reading (1987)
7. Srivastava, J., Cooley, R., Deshpande, M., Tan, P.: Web usage mining: Discovery and applications of usage patterns from web data. SIGKDD Explorations 1(2), 12–23 (2000)
8. Velayathan, G., Yamada, S.: Behavior-based web page evaluation. In: WI-IATW 2006: Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology, Washington, DC, USA, pp. 409–412. IEEE Computer Society, Los Alamitos (2006)

Journal Article Topic Detection Based on Semantic Features

Hei-Chia Wang[1], Tian-Hsiang Huang[1], Jiunn-Liang Guo[1,2], and Shu-Chuan Li[1]

[1] Institute of Information Management, University Road,
National Cheng Kung University, 701, Tainan, Taiwan
[2] Department of Aviation Management of R.O.C., Taiwan Air Force Academy,
Gangshan Township, Kaohsiung County 820, Taiwan
hcwang@mail.ncku.edu.tw, huangtx@gmail.com,
andyguo@cc.cafa.edu.tw, french127@hotmail.com

Abstract. The number of electronic journal articles is growing faster than ever before; information is generated faster than people can deal with it. In order to handle this problem, many electronic periodical databases have proposed keyword search methods to decrease the effort and time spent by users in searching the journal's archives. However, the users still have to deal with a huge number of search results. How to provide an efficient search, i.e., to present the search results in categories, has become an important current research issue. If search results can be classified and shown by their topics, users can find papers of interest quickly. However, traditional topic detection methods use only word frequencies, ignoring the importance of semantics. In addition, the bibliographic structures (e.g., **Title**, **Keyword**, and **Abstract)** have particular importance. Therefore, this paper describes a topic detection method based on bibliographic structures and semantic properties to extract important words and cluster the scholarly literature. The experimental results show that our method is better than the traditional method.

Keywords: Document clustering, Lexical chain, Semantic-based.

1 Introduction

The Internet today is going through a rapid phase of growth and development. With the growth of the Internet, information contained in electronic documents is increasingly widespread, with the World Wide Web as its primary repository. The convenience of electronic documents has motivated their more efficient application in information management and knowledge discovery. Given the avalanche of electronic documents, the pervasive use of search engines helps to minimize the time required to extract information.

In the most popular form of search, the search criteria are keywords, or concepts that may be contained in the electronic documents [1]. However, the users still have to deal with the overabundance of search results in some way. During the last decade, the question of how best to filter the results of search engines has become an important issue. Topic detection is an experimental method for automatically organizing search

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 644–652, 2009.

results. It could help users save time in identifying useful information from large scale electronic documents.

A topic is defined to be a seminal event or activity, along with all directly related events and activities [2]. Today, many different data mining methods are employed to recognize topics, for instance, the Naïve Bayes classifier [3], hierarchical clustering algorithms (HCA) [4-6], paragraph relationship maps [7], Formal Concept Analysis (FCA) [8] and lexicon chains [9-11]. These methods use the frequencies of words to calculate the similarity between two documents. Therefore, their accuracy is greatly hindered by the presence of synonyms. The present paper proposes a document topic detection method based on semantic features in order to improve the traditional method.

Halliday and Hasan [10] proposed a semantics-based method, the so-called lexical chain method, that can be used to identify the central theme of a document. Based on the lexical chain method, combined with the electronic Princeton WordNet lexical database (http://wordnet.princeton.edu/), the proposed method clusters documents by semantic similarity and extracts the important topics for each cluster. Ultimately, the method provides more user-friendly search results that are relevant to the topics.

Going one step further, one can note that full-text documents have bibliographic structures, such as **Title**, **Keyword**, and **Abstract**, each of specific importance. These structures should be considered in the clustering model. Alongside the universal electronic paper supply of journal databases, the targets for clustering are restricted to electronic papers from on-line journal databases that enable keyword search by users. We compared the performance of our proposed method with the traditional "term frequency with inverted document frequency" (TF-IDF) method [12]. The key contribution of this research is to design a method based on bibliographic structures and semantic properties to extract important words and cluster the literature. It can be used to retrieve topics and display the search results clustered by topics. Expert users can easily acquire literature of interest and correctly find information from the topic-cluster display.

2 Lexical Chain Construction

In order to exploit the semantic features to detect topics, we apply the following steps to construct lexical chains from a corpus [9, 11]:

1. Part-of-Speech tagging (POS) – Candidate words are nouns extracted from documents by POS tools.
2. Semantic comparison of nouns – Suppose that *A*, *B*, and *C* are nouns extracted that belong to the semantic sets $sense_A$, $sense_B$ and $sense_C$, respectively. When *A* matches with *B*, two kinds of permutation combinations are produced, as shown in Figure 1. The first permutation combination is $sense_A \cup sense_B$, which is presented as the lexical chain $chain_1$, and another permutation combination is $sense_A \cup \overline{sense_B}$, which is presented as $chain_2$, where $\overline{sense_B}$ is the set of senses of *B* that are not in $chain_1$. Furthermore, if *C* matches with *A* and *B*, there are four possible

permutation combinations: $chain_1 \cup sense_C$, $chain_1 \cup \overline{sense_c}$, $chain_2 \cup sense_C$, and $chain_2 \cup \overline{sense_c}$. According the rule to compare each noun until the whole document is processed.

3. Computation of link strength – As shown in Figure 1 (a), the link strength between *A* and *B* is 5, which means that the nearest same hypernym layer of *A* and *B* is 5.

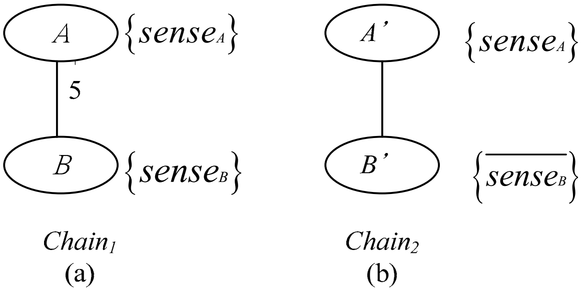


Fig. 1. Two permutation combinations (two lexical chains)

4. Repeat Step 2 and Step 3 recursively – Step 2 and Step 3 are repeated recursively until all the nouns have been processed.
5. Merge lexical chains – After the completion of the lexical chain construction, we check for lexical chains with the same semantics. If two lexical chains have same semantics, then they are merged.
6. Lexical chain selection – Setting the weight of a lexical chain to be its total link strength, we choose the lexical chain with the highest weight to represent the most important concept in a document.

3 Implementation of the Proposed Method

The proposed system architecture, depicted in Figure 2, is suggested to implement the proposed method. Note that CR_1 is the *1st* cluster, L_1 is the *1st* piece of literature, and the rest of the notation is analogous. There are three main models in the system architecture. At first, the pre-process model collects journal papers and processes their **Title**, **Keyword** and **Abstract** information to prepare for the lexical chain construction.

Secondly, the document representative model implements the steps described in Section 2 to build lexical chains. Figure 3 delineates the lexical chain building processes. Note that the WordNet is used to find the nearest hypernyms between two nouns. If the nearest layer hypernym is 1, then the link strength is given a weight of 10, and the second layer hypernym is given the weight 8. The rest is deduced by analogy, but if the hypernym is above the sixth layer, the weight is set to zero.

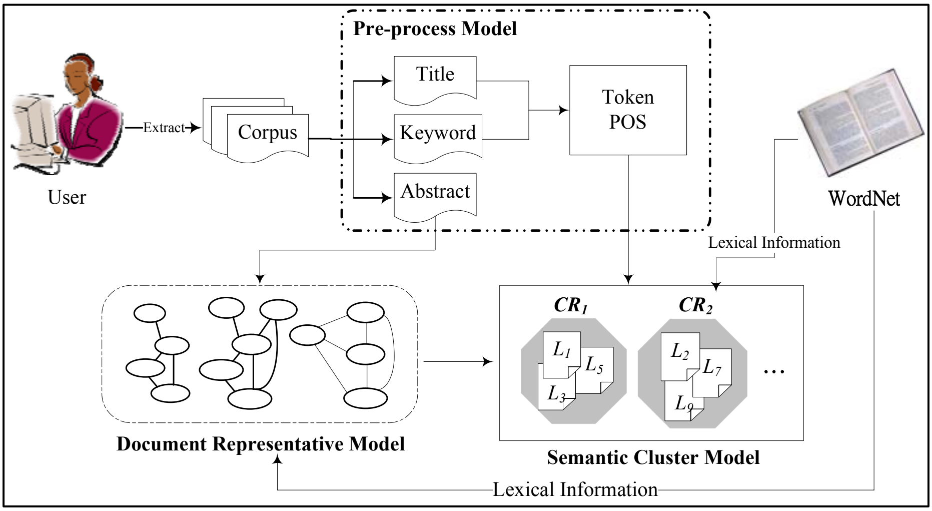


Fig. 2. The proposed system architecture

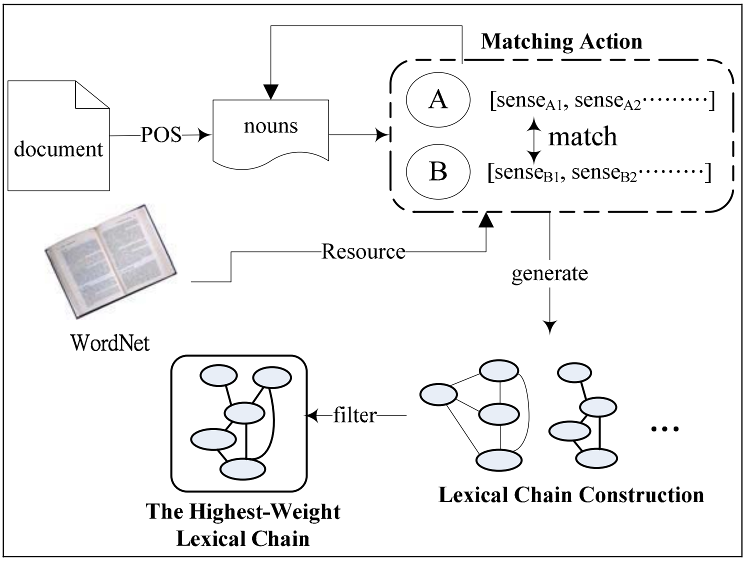

Matching Action
document
POS
nouns
A
[sense_{A1}, sense_{A2}·········]
match
B
[sense_{B1}, sense_{B2}·········]
WordNet
Resource
generate
filter
...
The Highest-Weight Lexical Chain
Lexical Chain Construction

Fig. 3. The lexical chain building processes

Formula (1) is used to calculate the link strength for two nouns:

$$relation(w_a, w_b) = \frac{weight_{w_a} + weight_{w_b}}{2}, \tag{1}$$

where $weight_{w_a}$ and $weight_{w_b}$ are the semantic weights of the nearest layer hypernyms between noun w_a and noun w_b in WordNet.

The tertiary model is the semantic cluster model. It calculates similarity and clusters documents. The most frequently used similarity measure in information retrieval is cosine similarity [13]. In order to take semantic features into account, in this paper we propose a novel method to calculate semantic similarity, in formulas (2) and (3). Note that similarities in the **Title**, **Keyword** and **Abstract** of papers will be calculated separately:

$$wf_i = 1 + \frac{WN_i}{N_i} \tag{2}$$

$$S_{ij} = \frac{\sum_{s=1}^{d} \sum_{g=1}^{h} (SemWe_{sg} \times wf_i \times wf_j)}{d \times h}, \tag{3}$$

where wf_i is the weight of the word frequency in document *i*, WN_i is the frequency of the word in document *i*, N is the total number of words in document *i*, *d* and *h* are the total numbers of all words in document *i* and *j*, respectively. *SemWe* is the weight that calculates the layer relation between word *s* and word *g* from hypernyms in WordNet. Then, according to pre-defined weights for **Title**, **Keyword** and **Abstract,** we adopted the total similarity score as the single-link distance [4, 6] between two different documents.

After the semantic similarity calculation, the HCA method is used to cluster the documents. Then, to take the calculation one step further, topics are extracted from documents. For example, suppose that a corpus has five documents, L_1, …, L_5. Figure 4 shows that L_1 and L_4 belong to one cluster, L_2 and L_5 belong to another cluster, and L_3 stands alone.

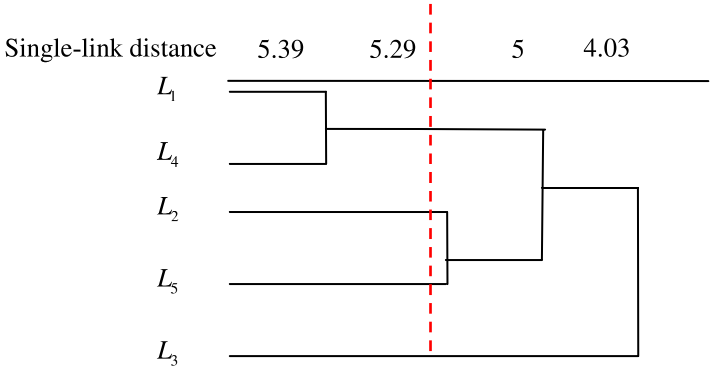


Fig. 4. An example of clustering documents by HCA

After collecting key phrases and their respective frequencies from each of the documents in one cluster, the key phrases with the highest frequencies can be viewed as topics of the cluster. Because users could not extrapolate easily from separate words, a Phrase Frequency (PF) method is proposed to extract topics. For example, suppose *A* (association rule), *C* (class association rule), *D* (data mining) and *S* (simple association rule) are four key phrases in a cluster CR_1. A is part of S and C. Hence, the PF of A is 3. The PFs of S and C are both 1. Figure 5 shows an example of the calculations. After sorting PF values, we extract the phrases with the top 20 percentages as topics. Ultimately, the topics of each cluster are extracted.

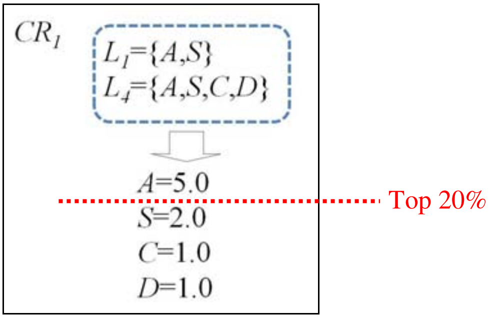


Fig. 5. An example of PF calculation

4 Performance Evaluation

Three datasets for performance evaluation were collected by searching the ISI and SDOS on-line databases by keywords, "Supply Chain Management," "Decision Support System," and "Data Mining." There are 100 documents in every dataset, and each document includes its **Title**, **Keyword** and **Abstract**.

The F-measure in formula (4) is used to evaluate performance of the proposed method compared to the TF-IDF method:

$$F-measure = \frac{2 \times precision \times recall}{precision + recall}, \tag{4}$$

where *precision* measures the ratio of correctly found documents over the total number of returned documents, and *recall* measures the ratio of correctly found documents over the total number of expected corresponding documents. Moreover, considering the bibliographic structure, five combinations of weights assigned to **Title**, **Keyword** and **Abstract** in the experiments are listed in Table 1. Figure 6 shows the results of the comparison between five combinations of weights of our method and the TF-IDF method. The F-measures of all five combinations are better than those of the TF-IDF method.

Table 1. Five combinations of weights assigned to **Title**, **Keyword** and **Abstract**

Combination	$w_{Abstract}$	w_{Title}	$w_{Keyword}$
1st	1	1	1
2nd	1	1	2
3rd	2	1	1
4th	1	2	1
5th	0.33	0.32	0.35

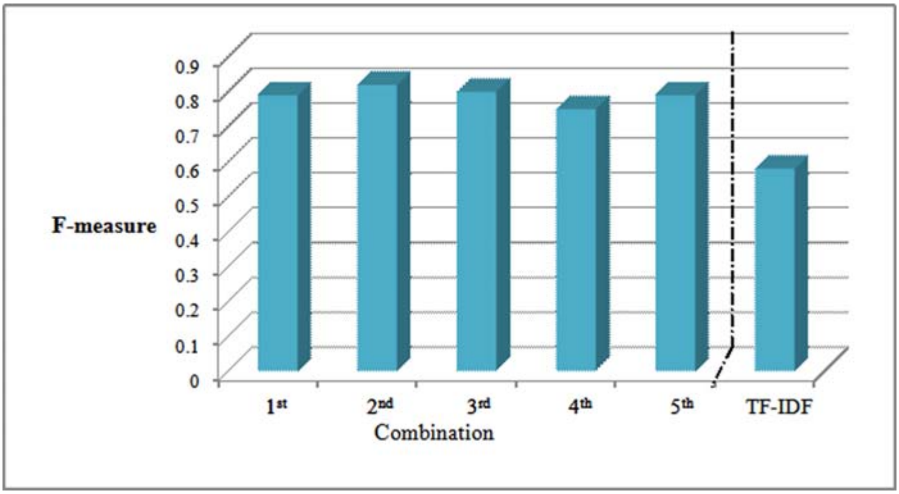


Fig. 6. Comparison results of the proposed method with the TF-IDF method

Furthermore, we consider that the bibliographic structures (**Title**, **Keyword**, and **Abstract)** may have different levels of importance for different types of documents. In order to explore whether a significant difference exists between **Title**, **Keyword** and **Abstract**, the statistical T-test was used with a significance level of 0.05. Ten independent tests were implemented, each with a document sample size of thirty. After similarity calculations and clustering, Table 2 shows the F-measure for each independent test. The F-measure of **Abstract** is significantly lower than the others, on average.

Table 2. The F-measure for each independent test

	Keyword	**Abstract**	**Title**
1st test	0.605050505	0.404411765	0.529100529
2nd test	0.707730866	0.351897352	0.603174603
3rd test	0.714889789	0.508258476	0.637555427
4th test	0.296650718	0.386161616	0.865319865
5th test	0.606871389	0.413822435	0.563530037
6th test	0.584415584	0.348355348	0.563530037
7th test	0.296650718	0.348355348	0.523809524
8th test	0.623824451	0.348484848	0.482146249
9th test	0.780219780	0.455433455	0.610158730
10th test	0.932659933	0.399637400	0.671957672
Average	0.614896373	0.396481804	0.605028267
Standard Deviation	0.187257352	0.050371778	0.102090517

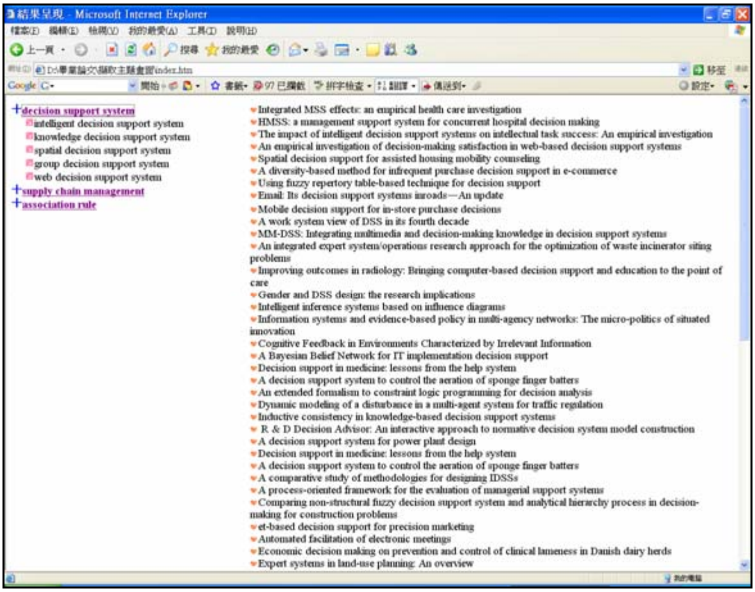
In addition, Table 3 shows the T-test results. These results demonstrate that **Keyword** and **Title** are of equal importance, and both more important than **Abstract**. Therefore, in future applications, we should assign the same weights to **Keyword** and **Title**, but lower weights to **Abstract**.

Table 3. The T-test results

	Hypothesis	T value	Testing result
Keyword versus **Title**	$H_0: \mu_{Keyword} \le \mu_{Title}$ $H_1: \mu_{Keyword} > \mu_{Title}$	0.14	Accept H_0
Keyword versus **Abstract**	$H_0: \mu_{Keyword} \le \mu_{Abstract}$ $H_1: \mu_{Keyword} > \mu_{Abstract}$	3.369	Reject H_0
Title versus **Abstract**	$H_0: \mu_{Title} \le \mu_{Abstract}$ $H_1: \mu_{Title} > \mu_{Abstract}$	5.5	Reject H_0

5 An Example of the System Interface

Figure 7 is an example of the system interface. The topics and sub-topics are shown on the left-hand side, and the documents are shown on the right-hand side. Users can look at documents by clicking the topics. The interface is used to help users find correct information using the topic-cluster display.


結果呈現 - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
搜尋
我的最愛
D:\畢業論文\擷取主題\index.htm
移至
Google

+decision support system
- intelligent decision support system
- knowledge decision support system
- spatial decision support system
- group decision support system
- web decision support system

+supply chain management
+association rule

- Integrated MSS effects: an empirical health care investigation
- HMSS: a management support system for concurrent hospital decision making
- The impact of intelligent decision support systems on intellectual task success: An empirical investigation
- An empirical investigation of decision-making satisfaction in web-based decision support systems
- Spatial decision support for assisted housing mobility counseling
- A diversity-based method for infrequent purchase decision support in e-commerce
- Using fuzzy repertory table-based technique for decision support
- Email: Its decision support systems inroads—An update
- Mobile decision support for in-store purchase decisions
- A work system view of DSS in its fourth decade
- MM-DSS: Integrating multimedia and decision-making knowledge in decision support systems
- An integrated expert system/operations research approach for the optimization of waste incinerator siting problems
- Improving outcomes in radiology: Bringing computer-based decision support and education to the point of care
- Gender and DSS design: the research implications
- Intelligent inference systems based on influence diagrams
- Information systems and evidence-based policy in multi-agency networks: The micro-politics of situated innovation
- Cognitive Feedback in Environments Characterized by Irrelevant Information
- A Bayesian Belief Network for IT implementation decision support
- Decision support in medicine: lessons from the help system
- A decision support system to control the aeration of sponge finger batters
- An extended formalism to constraint logic programming for decision analysis
- Dynamic modeling of a disturbance in a multi-agent system for traffic regulation
- Inductive consistency in knowledge-based decision support systems
- R & D Decision Advisor: An interactive approach to normative decision system model construction
- A decision support system for power plant design
- Decision support in medicine: lessons from the help system
- A decision support system to control the aeration of sponge finger batters
- A comparative study of methodologies for designing IDSSs
- A process-oriented framework for the evaluation of managerial support systems
- Comparing non-structural fuzzy decision support system and analytical hierarchy process in decision-making for construction problems
- et-based decision support for precision marketing
- Automated facilitation of electronic meetings
- Economic decision making on prevention and control of clinical lameness in Danish dairy herds
- Expert systems in land-use planning: An overview

Fig. 7. An example of the system interface

6 Conclusion and Future Work

In this paper, the lexical chain method is implemented to extract important concepts from documents. Then the documents are clustered by semantic similarity. The WordNet database is the lexicon used. In addition, the bibliographic structures **Title**, **Keyword** and **Abstract** are taken into consideration, with different weights given to each. We propose that **Keyword** and **Title** should have the same weight and that **Abstract** should have a lower weight**.** The experimental results show that the proposed method is better than the traditional TF-IDF method. Its key contribution is the ability to extract topics by semantic features, taking into account the influence of bibliographic structures, and to recommend clusters to users. However, a larger number of clusters may lead to the duplication of topics. This is a question that will require further attention in the future. In addition, syntactical analysis and phrase analysis may be used to increase the accuracy of the prediction, which are also suggestions for future research.

Acknowledgments. The research is based on work supported by 97-EC-17-A-05-S1-103 project from the Department of Industrial Technology, Ministry of Economic Affairs, Taiwan. We greatly benefited from the substantial support of our colleagues and students at the Web Knowledge Discovery Laboratory and Electric Motor Technology Research Center, National Cheng Kung University.

References

1. Voorhees, E.M.: Natural language processing and information retrieval. Springer, New York (1999)
2. Allan, J. (ed.): Topic detection and tracking: event-based information organization. Kluwer Academic Publishers, Dordrecht (2002)
3. Lee, M., Wang, W., Yu, H.: Exploring supervised and unsupervised methods to detect topics in biomedical text. BMC Bioinformatics 7 (2006)
4. Berkhin, P.: Survey of clustering data mining techniques. Accrue Sotware. Inc. (2002)
5. Hotho, A., Nürnberger, A., Paaß, G.: A brief survey of text mining. LDV Forum - GLDV Journal for Computational Linguistics and Language Technology (2005)
6. Kantardzic, M.: Data mining. Wiley Inter-Science, Hoboken (2003)
7. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
8. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Reidel, Dordrecht (1982)
9. Barzilay, R., Elhadad, M.: Using lexical chains for text summarization. In: ACL/EACL Workshop on Intelligent Scalable Text Summarization (1997)
10. Halliday, M., Hasan, R.: Cohesion in English. Longman, London (1976)
11. Hatch, P., Stokes, N., Carthy, J.: Topic detection, a new application for lexical chaining? In: The Proceedings of BCS- IRSG 2000, the 22nd Annual Colloquim on Information Retrieval Research, Cambridge, pp. 94–103 (2000)
12. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing & Management 24, 513–523 (1988)
13. Salton, G.: Automatic information organization and retrieval. McGraw-Hill, New York (1968)

Present Meets Past: Analysis of Internet Archive Quality

Dariusz Król[1] and Krzysztof Litwin[2]

[1] Institute of Informatics, Wrocław University of Technology,
Wybrzeże S. Wyspiańskiego 27, 50-370 Wrocław, Poland
dariusz.krol@pwr.wroc.pl
[2] Smart Technology Research Centre, Bournemouth University,
Poole, Dorset, BH12 5BB, UK
klitwin@bournemouth.ac.uk

Abstract. Web archives are powerful sources that could be used in many projects, to help improve the web quality. In this paper, we assess the quality and the capabilities of using the Internet Archive in domain-specific development. We implemented conference calendarium, the web system that provides the functionality of retrieving archived information about scientific conferences. The system was tested on ACM and IEEE conference portals. We analysed conference domain ontology regarding development of our system. Finally, we conclude that utilization of the past web pages can significantly improve quality of searching and browsing.

Keywords: Web archive, past web pages, Internet Archive, data quality.

1 Introduction

The effective measurement of web pages quality is crucial for today's Internet. The enormous number of sites covering almost every topic is very confusing to every net voyager who is seeking for information in this cavernous web. Hopefully search engines like Google's one comes forward to help us find the right way. Nowadays web pages quality measurement standards are very efficient, but before this what started only few years ago, the situation wasn't so bright. Of course the web was smaller, but big enough to bring a lot of difficulties while searching it. The Google's search algorithm was a mile stone in the Internet search engines business. But it always can be better, if we could increase the precision of PageRank algorithm or even include the possibility of future quality prediction.

The Semantic Web is an effective infrastructure to enhance visibility of knowledge on the web; it is the extension of the WWW that enables people to share content beyond the boundaries of applications and websites. The Semantic Web is about language for describing how the data relates to real world objects. It is also mainly concentrated on the interchange of documents, using common formats for integration and combination of data drawn from diverse sources. Like the Internet, the Semantic Web will be as decentralized as possible. Maybe idea of total consistency of its interconnections can be valuable for the users, but unchecked exponential growth will

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 653–662, 2009.

give us much more advantages, what the Internet has proved to us. Such compromise is a great benefit for the development of the Semantic Web. The core of the Semantic Web is ontology, which is used to explicitly represent our conceptualisations. The ontology engineering is supported by languages such as RDF, RDFS and OWL.

If we have enough of historical information we could predict how the page will act in the future based on this information [1]. Depending not only on current data, but also looking back we can much more reliably estimate the actual position of a page in PageRank and even predict how it would change during the future time. Very comfortable and vast source of data are web archive servers such as the Internet Archive with its Wayback Machine [5], which has 2 petabytes of data, 85 billion pages and still grows. Hence we have at our disposal enormous web pages library, updated every day [6, 8].

Web archives are powerful databases that could be used in many web projects, to help improve the web quality [2]. The goal of this work is the assessment of the Internet Archive quality in domain-specific development. With the knowledge about web archive capabilities we would know where it could be used and be aware of what future projects based on web archives would be made. Browsing past content of active web pages can help characterize them better. Today search engines are highly developed and very effective, but still improvements and fresh look on the subject is needed. Effective and accurate estimation of web pages quality is the most important subject from the point of view of Internet user. The comfort, the speed and data accessibility depends on the development of this subject.

The paper is organized as follows. In Section 2, we characterise web archives in general and the Internet Archive with its Wayback Machine in particular. In Section 3 the example of calendarium, domain-dependent quality evaluation of the Internet Archive was presented. We briefly discuss quality issues by presenting some experimental results. We conclude the paper in Section 4.

2 The Internet Archive and the Wayback Machine

The size and diversity of the web shows how easy it is to produce web content. Incredible amounts of data are published on the web every day, from various sources such as individuals, organizations or businesses. Also a lot of web pages disappear during the time. It is caused by many different reasons, such as: abandoned due to lack of interest or money; moved to different locations; hardware failures; viruses. Very often in undesired circumstances backups cannot be found. When Internet developers, users and various institutions recognized the need to store and preserve past content of the web for future use many archives came into being, and still many emerge. It is necessary to store the past content of web pages for future reuse, because the World Wide Web changes continuously, and unfortunately mostly in an unpredictable and unrecognized way. Specifically, a website can be suddenly lost without backup and can be reconstituted by probing web archives and search engine caches for missing content. Nowadays there are numerous archives containing snapshots of pages from many domains. Broadly we can specify three types of archives:

- General archives, containing pages covering almost every domain such as the Internet Archive.
- Dedicated archives, to which belong web pages from individual countries, for example Australia's web archive or Austrian archive and thematic web archives related to certain events or topics such as September 11 web archive or Election 2002 web archive.
- Pages repositories, for example local caches, personal web document repositories, transaction-time servers, site archives or search engine caches.

There are also organizations like International Internet Preservation Consortium that were established with a mission of collecting, preserving and making accessible data from web for future generations.

Web archives show the past states of societies and the history of the web. They show how our society has evolved and how it reacted to events in the past. They can potentially reveal the history of the underlying elements represented by archived pages. Starting from earliest snapshots we can also observe how technologies were evolving from simple to such sophisticated which are today.

Browsing the past web is currently not efficient due to three problems in particular:

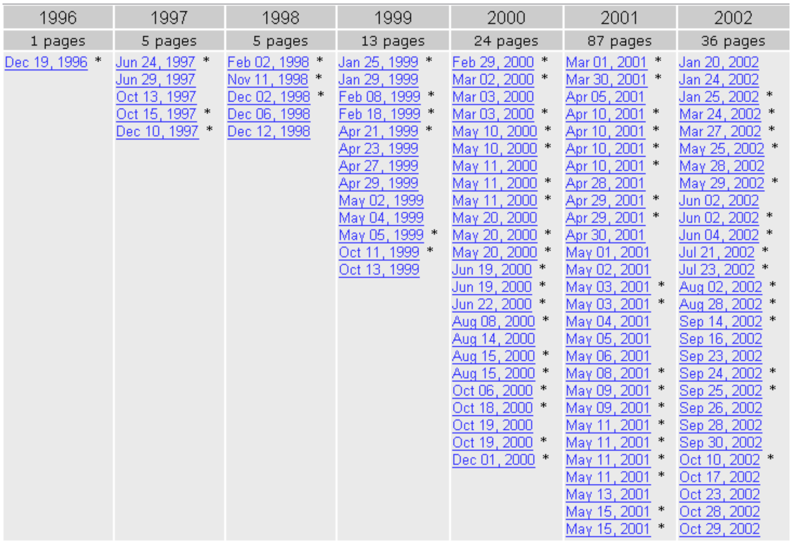
- Difficulty of navigation in space and time in web archives.
- Lack of efficient change management, which would help to perceive the evolution of pages in time and help to find required information in massive amounts of past web data.
- Limited coverage of web archives, hence, lack of historical data of many web pages or its incompleteness.

A major problem encountered when browsing web archives is an incomplete state of past web snapshots. Due to resource limitations, web archives cannot crawl and preserve the whole web with suitable granularity to capture all changes in web documents.

Recognizing the ephemeral and volatile nature of the web the Internet Archive has began to gather as much of the web as possible for future generations. It was founded in 1996 by Brewster Kahle with the intent of preserving what is arguably the fastest growing archive of human expression ever created. The Library of Congress, and other analog equivalents, keep copies of the thousands of books published every year, helping to preserve our paper history, but the amount of content on the Internet has grown to dwarf that repository. Also, unlike books on the shelves of libraries, web pages are in a constant state of flux: the average age of a web site is only 19 months with the average page changing every 100 days. The amount of information created and lost every day is staggering. The current state of digital technology and the Internet makes it feasible for the archive to reach it has stated mission of universal access to human knowledge.

The archive does not use specialized, very expensive servers, but ordinary PCs. Although an average server response time is quite long, the archive meets the usability requirements. All this problems are caused by financial nature reasons but the Internet Archive authorities cope with this excellently.

Until now the web archiving community has mostly concentrated on storing and preserving web pages; relatively little attention has been paid to providing interfaces to the past web data. The Wayback Machine is a popular web-based interface to the Internet Archive. It is more a general retrieval system than a traditional browsing application. After a user inputs a URL and optionally dates defining a requested time period, the available page versions of the URL are displayed in sequential order in the directory page (see Fig. 1). The user can click on any page to view its contents. He can follow links from that page if the linked pages are also stored in the archive. The Wayback Machine also indicates which page versions contain changes by marking them with an asterisk; however, it does not display changes unless the user explicitly requests change visualization. Generally, the Wayback Machine does not promote easy serendipitous discovery in the Internet Archive. Horizontal browsing along the time line is troublesome and takes time since the user needs each time to come back to the directory page and to click on a new page version. This becomes even more difficult when user wants to freely browse the temporal structure of the past web.



1996	1997	1998	1999	2000	2001	2002
1 pages	5 pages	5 pages	13 pages	24 pages	87 pages	36 pages
Dec 19, 1996 *	Jun 24, 1997 *	Feb 02, 1998 *	Jan 25, 1999 *	Feb 29, 2000 *	Mar 01, 2001 *	Jan 20, 2002
	Jun 29, 1997	Nov 11, 1998 *	Jan 29, 1999	Mar 02, 2000 *	Mar 30, 2001 *	Jan 24, 2002
	Oct 13, 1997	Dec 02, 1998 *	Feb 08, 1999 *	Mar 03, 2000	Apr 05, 2001	Jan 25, 2002 *
	Oct 15, 1997 *	Dec 06, 1998	Feb 18, 1999 *	Mar 03, 2000 *	Apr 10, 2001 *	Mar 24, 2002 *
	Dec 10, 1997 *	Dec 12, 1998	Apr 21, 1999 *	May 10, 2000 *	Apr 10, 2001 *	Mar 27, 2002 *
			Apr 23, 1999	May 10, 2000 *	Apr 10, 2001 *	May 25, 2002 *
			Apr 27, 1999	May 11, 2000	Apr 10, 2001 *	May 28, 2002
			Apr 29, 1999	May 11, 2000 *	Apr 28, 2001	May 29, 2002 *
			May 02, 1999	May 11, 2000 *	Apr 29, 2001 *	Jun 02, 2002
			May 04, 1999	May 20, 2000	Apr 29, 2001 *	Jun 02, 2002 *
			May 05, 1999 *	May 20, 2000 *	Apr 30, 2001	Jun 04, 2002 *
			Oct 11, 1999 *	May 20, 2000 *	May 01, 2001	Jul 21, 2002 *
			Oct 13, 1999	Jun 19, 2000 *	May 02, 2001	Jul 23, 2002 *
				Jun 19, 2000 *	May 03, 2001 *	Aug 02, 2002 *
				Jun 22, 2000 *	May 03, 2001 *	Aug 28, 2002 *
				Aug 08, 2000 *	May 04, 2001	Sep 14, 2002 *
				Aug 14, 2000	May 05, 2001	Sep 16, 2002
				Aug 15, 2000 *	May 06, 2001	Sep 23, 2002
				Aug 15, 2000 *	May 08, 2001 *	Sep 24, 2002 *
				Oct 06, 2000 *	May 09, 2001 *	Sep 25, 2002 *
				Oct 18, 2000 *	May 09, 2001 *	Sep 26, 2002
				Oct 19, 2000	May 11, 2001 *	Sep 28, 2002
				Oct 19, 2000 *	May 11, 2001 *	Sep 30, 2002
				Dec 01, 2000 *	May 11, 2001 *	Oct 10, 2002 *
					May 11, 2001 *	Oct 17, 2002
					May 13, 2001	Oct 23, 2002
					May 15, 2001 *	Oct 28, 2002
					May 15, 2001 *	Oct 29, 2002

Fig. 1. List of ACM available snapshots from the period 1996–2002

Recently, a public open source implementation of Wayback Machine has been also released. Besides having most of the functionalities of the standard web-based version, it provides Timeline Replay mode in which a timeline listing page versions is visible at the top of each page. Browsing forward and backward in time is done by clicking on the timeline. However, again, his style of browsing requires users to click on page versions, which might be troublesome when there are many past page snapshots.

The browser for past web has been already implemented [3]. Two basic types of browsing have been distinguished: vertical and horizontal. Traditionally navigating past versions of pages means navigating between past versions of different pages at or around a certain point in time, while in this case means viewing page versions of the same page along the time direction, i.e. browsing the past web in a horizontal direction. A mixture of both kinds of browsing would enable users to traverse both in time and space of the past web. Proposed framework incorporates a mixture of both types.

There is no doubt that the Internet Archive is one of the largest or even the largest repository of web pages and media files with open and free access. Hence it was chosen for deeper analysis.

3 Calendarium – Domain-Dependent Evaluation of the Past

Ontologies have become common on the World Wide Web [4]. Ontologies on the web range from large taxonomies categorizing web sites to categorizations of products for sale and their features. Adding a semantic dimension to web pages can improve and simplify of information retrieval from the web. Internet enlarges exponentially, in consequence processing heterogeneous data sources and retrieving relevant data become very complex. Use of ontology allows:

- Sharing understanding of the structure of information.
- Enabling reuse of domain knowledge.
- Making domain assumptions explicit.
- Separating domain knowledge from the operational knowledge.
- Analysing domain knowledge.

Enabling reuse of domain knowledge and analysing domain knowledge are what we in particular use ontology for. A useful conference domain ontology has been open source published [7]. Below is presented main extract from its OWL.

```
<rdf:RDF>
   <owl:Class rdf:ID="Event"/>
   <owl:Class rdf:ID="Location"/>

<rdf:Description rdf:about="http://xmlns.com/foaf/0.1/Person">
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>

<rdf:Description rdf:about="http://xmlns.com/foaf/0.1/Organization">
   <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>

<owl:Class rdf:ID="Person">
   <owl:equivalentClass rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
</owl:Class>

<owl:Class rdf:ID="Group"/>

<owl:Class rdf:ID="Organization">
   <rdfs:subClassOf rdf:resource="http://xmlns.com/foaf/0.1/Organization"/>
   <rdfs:subClassOf rdf:resource="#Group"/>
</owl:Class>

<owl:ObjectProperty rdf:ID="memberOf">
   <rdfs:domain rdf:resource="#Person"/>
   <rdfs:range rdf:resource="#Group"/>
   <owl:inverseOf rdf:resource="#hasMember"/>
</owl:ObjectProperty>

<owl:DatatypeProperty rdf:ID="hasStartDate">
   <rdfs:domain rdf:resource="#Event"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#dateTime"/>
</owl:DatatypeProperty>
```

```
<owl:Class rdf:ID="Committee">
   <rdfs:subClassOf rdf:resource="#Group"/>
   <rdfs:subClassOf>
      <owl:Restriction>
         <owl:onProperty rdf:resource="#hasMember"/>
         <owl:minCardinality rdf:datatype="
http://www.w3.org/2001/XMLSchema#nonNegativeInteger"> 1
         </owl:minCardinality>
       </owl:Restriction>
      </rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="SocialEvent">
   <rdfs:subClassOf rdf:resource="#Event"/>
</owl:Class>
</rdf:RDF>
```

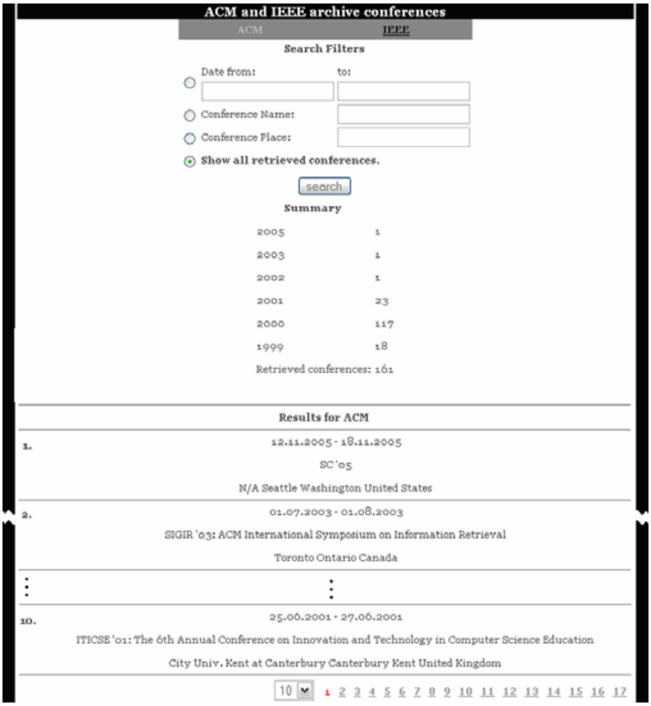
The ontology is composed of following classes: Event, Location, Person, Group, Organization, Committee, Panel, Conference, Workshop, Chair, CommitteeMember, Attendee, Presentation, Presenter, KeynoteSpeaker, KeynoteTalk, and SocialEvent. In this particular case this representation is too complex, it has elements that cannot be retrieved from the archive and consequently use of small representation containing only basic information about conference is more advisable.

To achieve the goal and test capabilities of the Internet Archive a web application has been developed. The web interface has been chosen because of its accessibility and maintainability. The first question was how get a list of desired snapshots. Unfortunately before the date when research was finished, there was no search option available. There were also no premises for including this option in the nearest future. The only method for retrieving desired data was to know the exact URL of the past resource. For systems constantly retrieving data of various origin it is a serious difficulty. The easiest but far from the satisfactory accuracy is a method of searching Google with given query and using received URL addresses in searching the past data.

Another problem is that page address could be changed and the present address retrieved from Google may not match the past one. A time address changes can affect with the significant data lost while searching relevant data. The past URL address is built by the following pattern: www.archive.org/snapshot_nr/page_past_address. If the snapshot_nr is equal to “*” than server will return page with all available snapshots, with their dates given as links, hence it is possible to search available archive pages by their addresses remotely. Having direct URL to chosen snapshot gives easy way to download page content.

Research was taken on the conference domain and three positions were chosen as objects of interest: ACM (www.acm.org), IEEE (www.ieee.org), and Springer (www.springer.com) portals. Springer was removed from the list because of the lack of archived data; fortunately two other portals had plenty of valuable snapshots. Consequently the application provides to the user functionality of retrieving archived information about conferences from two sources which are ACM and IEEE portals. After gathering as much information from archives as it was possible at this time, the next step was to retrieve all data from present versions of chosen portals. The user has to his disposal the set of data filters; it is possible to restrict the time scope of presented results, search by a conference name or a location. In the application has been also applied paging, for comfortable looking through the results. In the bottom of every page a diagram is available, that confronts numbers of conferences harvested from past and present web.

Proposed system was implemented as a web application with use of Java Enterprise Edition technology. Application flow is as follows: It downloads pages from archive, the localizations that should be checked are hard-coded and then retrieved HTML pages are being parsed. Gathered information is serialized to avoid excess data retrievals, it strongly enhances efficiency of the application. Daily at midnight information about conferences is automatically updated. Fig. 2 shows data presented by the system for ACM option. On the screenshot we can observe the summary of all conferences retrieved from the archive.



○ Date from: to:

○ Conference Name:

○ Conference Place:

● Show all retrieved conferences.

2005	1
2003	1
2002	1
2001	23
2000	117
1999	18

Retrieved conferences: 161

1. 12.11.2005 - 18.11.2005
SC '05
N/A Seattle Washington United States

2. 01.07.2003 - 01.08.2003
SIGIR '03: ACM International Symposium on Information Retrieval
Toronto Ontario Canada

⋮ ⋮

10. 25.06.2001 - 27.06.2001
ITICSE '01: The 6th Annual Conference on Innovation and Technology in Computer Science Education
City Univ. Kent at Canterbury Canterbury Kent United Kingdom

10 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

Fig. 2. Screenshot of the system presenting data retrieved from the ACM portal

The speed of archive servers was significantly low; servers were also returning 503 very often. ACM conferences were collected from only few pages and total time of the operation was just 18 seconds, whereas IEEE conferences where collected from many pages (each conference from separate page). The total operation time of the operation was over 28 minutes. We use Internet line 1 Mb/s fast. The IEEE portal caused a problem, because it was using JEE technology with Struts framework, but luckily web robots gather dynamic pages as well.

The list below presents problems that we have encountered:

- “Robots.txt” – these are scripts that attached to a page disallow crawlers taking a snapshot. If page owner will attach the script in the future after the snapshot was taken, archive authorities will block access to already taken snapshots, so this works backwards according to the Internet Archive.

- Failed connection – this is a message that you can receive while traversing the archive; it communicates problems with connecting to servers.
- Not in archive – it means that a web page that you are trying to reach was not archived, for example link in a snapshot can point a page that is not in the archive.
- No matches found – message shows when no matches for given URL in the archive were found.
- File location error – occurs when a server has encountered problems with physically localising the file.
- Path index error – the URL has been archived but not indexed.
- Incompleteness of pages – often content of archived pages is not archived fully. There are many lacks of data of different type. Mostly graphics were unavailable. If the page was older then more graphics were unavailable.
- Broken links – it is an often problem according to the Internet Archive, sometimes among big number of outbound links on a page; most of them were pointing to pages that were not saved.

The navigation in past versions of web sites among pages that those sites were pointing at that time is possible, because every link on the page is being changed into link pointing to the archival version of the page. Certainly this mechanism is far from being perfect and causes many anomalies on pages.

One of the biggest problems is "Robots.txt" script, which can be very troublesome. Even if there is enormous number of pages available in the archive, they often cannot be accessed in cause of those scripts. Owners of sites can include script like this in their page and stop the Internet Archive web crawlers before gathering the page. Additionally it works retroactively, it means that including script on the page will affect in blocking access to pages that were archived in the past.

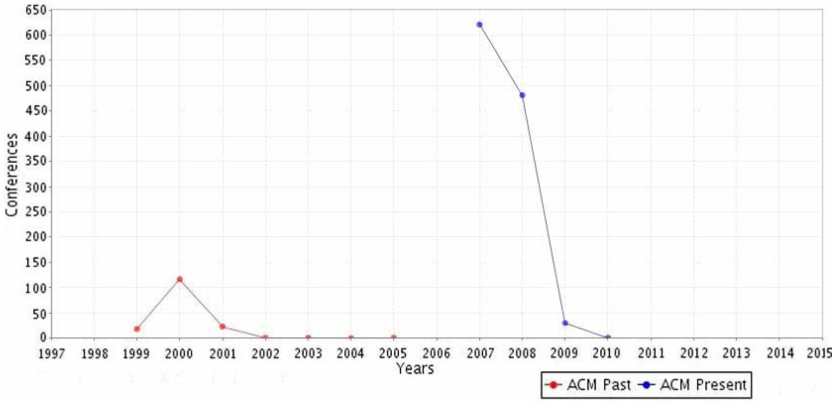


Fig. 3. Number of conferences retrieved from Past and Present of the ACM portal

Beside of anomalies that are only noticeable in archives, following anomalies (available in the present web) can be observed as well: defect in high level design, error in frame decomposition, navigation problem, missing translation, unintended language switch, internal/external hyperlink inconsistency, deviation from expected behaviour, duplicated pages.

Analysing Fig.3, with results obtained from the ACM portal, we can notice that conferences before year 2007 can be retrieved only from archives. It is a huge advantage of the archive collection. The number of archive conferences is not extensive but increases the total number of available information according to both resources simultaneously.

As we can read from Fig.4, with results obtained from the IEEE portal, number of retrieved conferences from actual portal is overwhelming. From the current version of the portal we can obtain only conferences that had place after 2003, whereas dates of conferences retrieved from past versions of the portal begin since 1998.

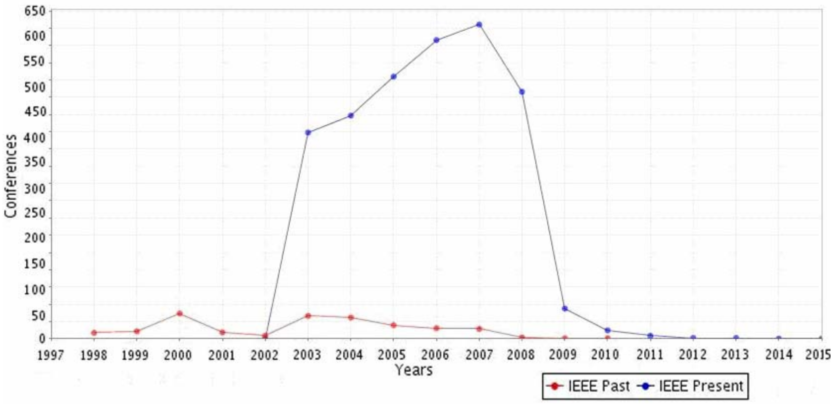


Fig. 4. Number of conferences retrieved from Past and Present of the IEEE portal

4 Conclusion

Web archives have been discussed; in particular the Internet Archive was taken into deeper analysis. Specifically the quality of the Internet Archive in conference domain approach was described. Tests were carried out on ACM and IEEE portals. Data gathering is a major problem of almost all evaluation systems. Ontologies can create an effective infrastructure that enhances visibility of the web knowledge; it is the extension of the World Wide Web that enables finding and retrieving in easy way valuable information. Ontologies are commonly described in OWL language, which is very comfortable in use and provides an efficient tool for data processing.

It is important that Internet developers eventually notice that utilization of the past web can significantly improve searching and browsing. Focusing only on present versions of pages deprives developers of massive, valuable amount of data. This type of attitude makes it impossible to get to know examined page in such wide scope, which provides combination of “past” and “present approaches”. Web archives show the history and evolution of the web. When web authors may gain more knowledge about certain page, it helps them in making a decision about linking or not linking to that page. With knowledge about archive pages we can tune up known web quality with a retrospective aspect. Looking back can show us the future of a page; it means we are able to predict how the page will act in the future basing on its historical information. For many users PageRank position is the most important. PageRank with included retrospective aspect can much more reliably estimate the current position of a page and even predict how it would change during the future time. Such enormous repositories of historical information cannot be ignored; there are many possibilities of utilising this data. Past information is often incomplete or resource that we are looking for is not archived, but disadvantages of this type with no doubt will not prevail over benefits for using it.

Also a little survey of www.bbc.co.uk archived websites was taken. It is a very wide set of snapshots, that reaches valuable data since 1997. We can browse first pages of the site through the time and have a look what articles have been published and get to know events from a current date. Many articles have not been archived or have been blocked. It is a future work to be done, to examine the amount of valuable data that can be retrieved from this domain.

Future work includes developing efficient mechanism for certain data retrieval from archive servers. Nowadays it is very difficult to localise data from certain domain, thus attempt of creating method for such retrieval should to be taken. Having this tool we can much more easily develop sophisticated application for efficient past web utilization.

References

1. Cao, Y., Lim, E.P., Ng, W.K.: Data model for warehousing historical Web information. Information & Software Technology 45(6), 315–334 (2003)
2. Jatowt, A., Kawai, Y., Ohshima, H., Tanaka, K.: What Can History Tell Us? Towards Different Models of Interaction with Document Histories. In: Proceedings of the 19th ACM Conference on Hypertext and Hypermedia, pp. 5–14. ACM, New York (2008)
3. Jatowt, A., Kawai, Y., Nakamura, S., Kidawara, Y., Tanaka, K.: Journey to the Past: Proposal of a Framework for Past Web Browser. In: Proceedings of the 17th ACM Conference on Hypertext and Hypermedia, pp. 135–144. ACM, New York (2006)
4. Jurisica, I., Mylopoulos, J., Yu, E.S.K.: Ontologies for Knowledge Management: An Information Systems Perspective. Knowl. Inf. Syst. 6(4), 380-401 (2004)
5. Lee, M., Cho, S., Choi, E.: Web Archive System for Efficient Storage of Web History Information. In: Proceedings of the 2007 international Conference on Intelligent Pervasive Computing, pp. 378–381. IEEE Computer Society, Los Alamitos (2007)
6. Martin, K.E., Eubank, K.: The North Carolina State Government Website Archives: A case study of an American government Web archiving project. New Review of Hypermedia and Multimedia 13(1), 7–26 (2007)
7. SchemaWeb - RDF Schemas Directory, `http://www.schemaweb.info` [01.12.2008]
8. Žabika, P., Matjka, l.: Czech Web archive analysis. New Review of Hypermedia and Multimedia 13(1), 27–37 (2007)

Constructing Decisional DNA on Renewable Energy: A Case Study

Cesar Sanin and Edward Szczerbicki

The University of Newcastle, Faculty of Engineering and Built Environment,
School of Mechanical Engineering,
University Drive, Callaghan, NSW 2308
{Cesar.Sanin,Edward.Szczerbicki}@newscastle.edu.au

Abstract. Knowledge engineering techniques are becoming useful and popular components of hybrid integrated systems used to solve complicated practical problems in different fields. Knowledge engineering techniques offer the following features: learning from experience; handling noisy and incomplete data; dealing with non-linear problems; and predicting. This paper presents a knowledge engineering case study by constructing a chromosome of Energy Decisional DNA. Decisional DNA, as a knowledge representation structure, offers great possibilities on gathering explicit knowledge of formal decision events as well as a tool for decision making processes. In this case study, several Sets of Experience of geothermal energy were collected for the construction of a geothermal chromosome within the Energy Decisional DNA. This chromosome is then implemented in an ontology model aiming to be used for predicting purposes. Thus, it enhances different systems with predicting capabilities and facilitates knowledge engineering processes inside decision making.

Keywords: Knowledge Engineering, Decisional DNA, Knowledge Representation, Renewable Energy, Set of Experience Knowledge Structure, Decision Making.

1 Introduction

The term knowledge engineering has been defined as a discipline that aims to offering solutions for complex problems by the means of integrating knowledge into computer systems [7]. It involves the use and application of several computer science domains such as artificial intelligence, knowledge representation, databases, and decision support systems, among others. Knowledge engineering technologies make use of the synergism of hybrid systems to produce better, powerful, more efficient and effective computer systems.

Among the features associated with knowledge engineering systems are human intelligence capabilities such as learning, interpolation and forecasting from current knowledge or experience. In our case, experience is the main and most appropriate source of knowledge and its use leads to useful systems with improved performance. Knowledge engineering techniques have been used in several domains and applications and the energy field is not an exception [4], [5], [10], [11], [12].

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 663–671, 2009.

Renewable energy (RE) resources have great potential and can supply the present world energy demand. RE can reduce atmospheric emissions, offer diversity in energy supply markets, and guarantee long-term sustainable energy supply [2]. In addition, the use of RE resources has been outlined as a research focus in many developed countries.

One promising RE technology is geothermal energy. A geothermal system's performance, i.e. it can work properly, efficiently and economically to meet the desired load requirements under local ground conditions, depends upon several factors, including ambient and ground temperatures, pressures, and ground matter, etc. In any geothermal system, sizing represents an important part of the system design, e.g., the optimal pressure or the temperatures associated with those pressures which determine the size of certain components. Thus, in order to size a geothermal system, the characteristic performance of each component in the system is required.

The present paper explains the process of implementing a knowledge engineering technology that can collect, use and offer trustable knowledge in the geothermal energy domain. We make use of the Set of Experience (SOE) as a knowledge structure that allows the acquisition and storage of formal decision events in a knowledge-explicit form. It comprises variables, functions, constraints and rules associated in a DNA shape allowing the construction of the Decisional DNA of an organization. Having a powerful knowledge structure such as the SOEKS in the Decisional DNA and apply it to the construction of a geothermal energy chromosome can enrich and develop RE research.

2 Background

2.1 Set of Experience Knowledge Structure (SOEKS) and Decisional DNA

Arnold and Bowie [1] argue that "the mind's mechanism for storing and retrieving knowledge is transparent to us. When we 'memorize' an orange, we simply examine it, think about it for a while, and perhaps eat it. Somehow, during this process, all the essential qualities of the orange are stored [experience]. Later, when someone mentions the word 'orange', our senses are activated from within [query], and we see, smell, touch, and taste the orange all over again". The SOEKS has been developed to keep formal decision events in an explicit way [14]. It is a model based upon existing and available knowledge, which must adjust to the decision event it is built from (i.e. it is a dynamic structure that relies on the information offered by a formal decision event); besides, it can be expressed in OWL as an ontology in order to make it shareable and transportable [15], [16]. Four basic components surround decision-making events, and are stored in a combined dynamic structure that comprises the SOE; they are: *variables V*, *functions F*, *constraints C*, and *rules R*.

Additionally, the SOEKS is organized taking into account some important features of DNA. Firstly, the combination of the four nucleotides of DNA gives uniqueness to itself, just as the combination of the four components of the SOE offer distinctiveness. Moreover, the elements of the structure are connected among themselves imitating part of a long strand of DNA, that is, a gene. Thus, a gene can be assimilated to a SOE, and in the same way as a gene produces a phenotype, a SOE produces a value of decision in terms of the elements it contains. Such value of decision can be called the efficiency or the phenotype value of the SOE [14]; in other words, the SOEKS, itself, stores an answer to a query presented.

A unique SOE cannot rule a whole system, even in a specific area or category. Therefore, more Sets of Experience should be acquired and constructed. The day-to-day operation provides many decisions, and the result of this is a collection of many different SOE. A group of SOE of the same category comprises a decisional chromosome, as DNA does with genes. This decisional chromosome stores decisional "strategies" for a category. In this case, each module of chromosomes forms an entire inference tool, and provides a schematic view for knowledge inside an organization. Subsequently, having a diverse group of SOE chromosomes is like having the Decisional DNA of an organization, because what has been collected is a series of inference strategies related to such enterprise (Fig. 1).

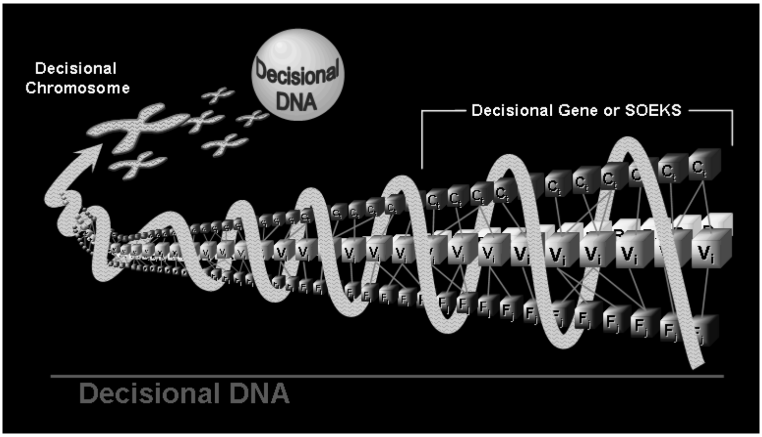


Fig. 1. SOEKS and Decisional DNA

In conclusion, the SOEKS is a compound of variables, functions, constraints and rules, which are uniquely combined to represent a formal decision event. Multiple SOE can be collected, classified, and organized according to their efficiency, grouping them into decisional chromosomes. Chromosomes are groups of SOE that can accumulate decisional strategies for a specific area of an organization. Finally, sets of chromosomes comprise what is called the Decisional DNA of the organization [13].

2.2 The Geothermal Laboratory

The University of Newcastle counts with a geothermal laboratory controlled by the Geothermal Research Group (GRG) on a project sponsored by Granite Power and AusIndustry through the Renewable Energy Development Initiative (REDI) grant. The GRG develops several experiments aiming for efficient and optimal production of power cycles for generation of electricity from geothermal reservoirs [13].

The geothermal laboratory simulates traditional geothermal power cycles in which a working fluid is used to convert heat energy into mechanical energy (fig. 2). The pressure of the working fluid is increased in the Compression Unit and passed on to the Heat Generator (1) where the fluid temperature is raised by the waste heat from a latter part of the cycle. This partially heated fluid (1') is then passed through the High Temperature Heat Exchanger where heat exchange with the geothermal fluid will raise the working fluid temperature to its maximum. The hot working fluid (2) then passes through the Turbine, where its expansion forces the rotation of the turbine that drives an electrical generator. This expansion step reduces the temperature and pressure of the fluid. The exiting stream from the turbine (3) passes through the Heat Generator where energy is recovered and injected into the cooler compressed fluid (1). Following the Heat Generator, a separate Cold Fluid in the Low Temperature Heat Exchanger cools the working fluid (3') to its lowest temperature. The cold, working fluid (4) now re-enters the Compression Unit to be cycled once again [13].

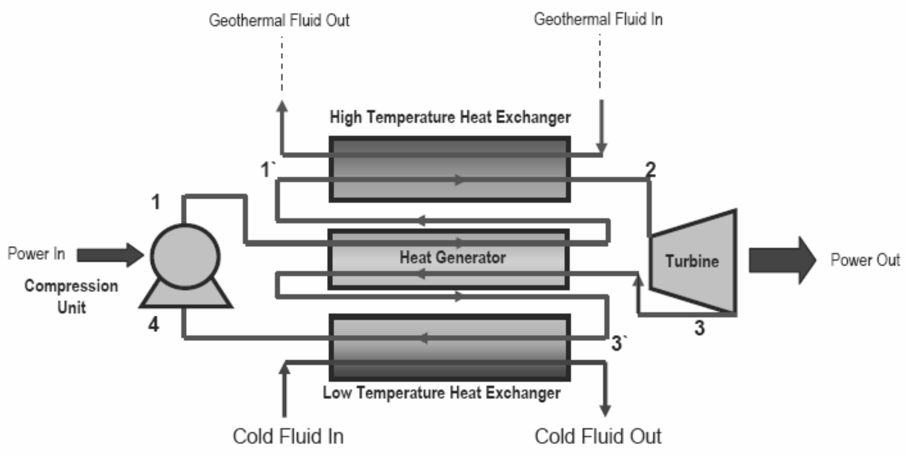


Fig. 2. Geothermal power cycle flow diagram [13]

In order to collect the research knowledge and experience created for this geothermal power cycle laboratory, several instruments and sensors are positioned in the system.. Due to the experimental nature of the work, the geothermal system incorporates a state of the art digital control system to continuously monitor, control, and record all process parameters. The primary roles of the digital control system are monitoring the process and collect online data from all the process sensors. Additionally, the digital control systems acts as a safe-fail device that controls the process and safely shut down the system in the event of a safety breach. The electronic system was supplied by Emerson Process Management Pty. Ltd. and consists of: a PC based control centre, a Foundation Field bus digital control system (DeltaV), 16 pressure transmitters, 24 temperature transmitters and 3 flow meters (a section of the laboratory is shown in fig. 3). The PC is connected to the field instruments via the DeltaV, digital control system.

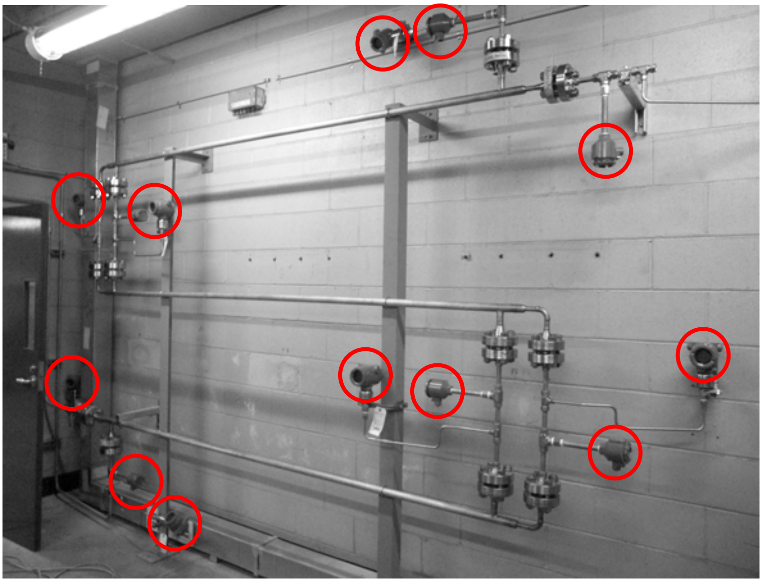


Fig. 3. Section of the geothermal laboratory with process sensors

The PC based remote operator control panel operates the Emerson PlantWeb software, which provides a visual user interface. The PC is connected to the field instruments via the DeltaV and collects all the real time online information of the process sensors. This is the repository of information.

3 Chromosome on Geothermal Energy

Experimental Research produces myriads of information and data which is collected and, sometimes, analyzed and stored; however, after a while, this information is commonly disregarded, not shared, and put behind. Little of this collected information survives, and in some cases, over time, it becomes inaccessible due to poor knowledge management practices or technology changes such as different software, hardware or storage media [3], [6], [8], [9]. These events indicate that there is a clear deficiency on information storage of experimental research, resulting in lost knowledge of the experimental processes. We suggest that some of the reasons could be a lack of technologies, not encouragement of the academic environment, and additional work involved on storing such experience. Through our project we proposed three important elements: *(i)* a knowledge structure able to store and maintain experiential knowledge, *(ii)* a solution for collecting experience that can be applied to experimental research, and *(iii)* a way to automate decision making by using such experience. In this case study we apply the three elements on RE.

Our plan aimed for the construction of a chromosome on geothermal energy which involved the implementation of the Decisional DNA, and within it, the use of the

SOEKS, to store and maintain experimental experience. Such plan requires different steps, they are as follows:

a) Run experiments at the geothermal laboratory,
b) Gather online information from the 43 sensors into the DeltaV,
c) Convert information to SOEKS compliant XML (eXtensible Markup language) by parsing the information onto metadata,
d) Feed an ontology chromosome with Sets of Experience, and
e) Check Energy Decisional DNA by querying the system with samples.

At the geothermal laboratory, experiments can run continuously for about 20 hours and information from each sensor can be collected at defined intervals between 1 second and 1 hour. For testing purposes, samples from two different experiments involving 2 fluids at intervals of 1 minute are sufficient. Such experiments produced 2400 events, each one involving variables of 43 sensors at a time. This information collected by fieldbus technology arrives to the DeltaV and it is stored in comma separated values (CSV) files.

Having these files, a parser written in Java, built up the XML trees which are then uploaded by using a Java API into protégé (an ontology software used for the construction of the Decisional DNA). Once the geothermal chromosome is constructed and ready in protégé, an additional extension for ontologies was applied: *Reflexive Ontologies*.

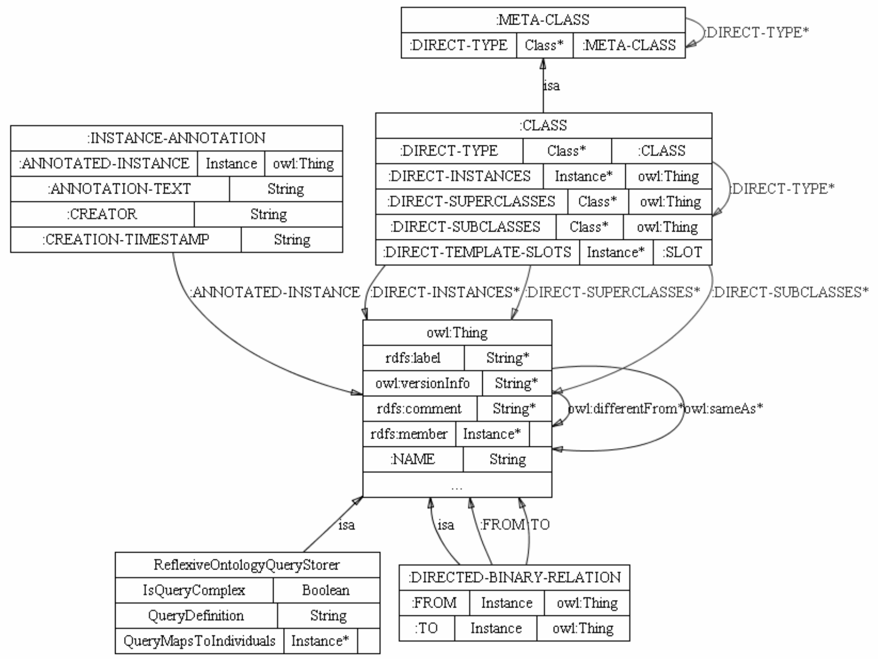

:META-CLASS
:DIRECT-TYPE | Class* | :META-CLASS
:DIRECT-TYPE*
isa
:CLASS
:DIRECT-TYPE | Class* | :CLASS
:DIRECT-INSTANCES | Instance* | owl:Thing
:DIRECT-SUPERCLASSES | Class* | owl:Thing
:DIRECT-SUBCLASSES | Class* | owl:Thing
:DIRECT-TEMPLATE-SLOTS | Instance* | :SLOT
:DIRECT-TYPE*
:INSTANCE-ANNOTATION
:ANNOTATED-INSTANCE | Instance | owl:Thing
:ANNOTATION-TEXT | String
:CREATOR | String
:CREATION-TIMESTAMP | String
:ANNOTATED-INSTANCE
:DIRECT-INSTANCES*
:DIRECT-SUPERCLASSES*
:DIRECT-SUBCLASSES*
owl:Thing
rdfs:label | String*
owl:versionInfo | String*
rdfs:comment | String*
rdfs:member | Instance*
:NAME | String
...
owl:differentFrom*
owl:sameAs*
isa
isa
:FROM
:TO
ReflexiveOntologyQueryStorer
IsQueryComplex | Boolean
QueryDefinition | String
QueryMapsToIndividuals | Instance*
:DIRECTED-BINARY-RELATION
:FROM | Instance | owl:Thing
:TO | Instance | owl:Thing

Fig. 4. Reflexive Ontology class schema

Reflexivity addresses the property of an abstract structure of a knowledge base (in this case, an ontology and its instances) to "know about itself". When an abstract knowledge structure such as the SOEKS is able to maintain, in a persistent manner, every query performed on it, and store those queries as individuals of a class that extends the original ontology, it is said that such ontology is reflexive, i.e. a reflexive ontology (RO). Therefore, any RO is an abstract knowledge structure with a set of structured contents and relationships, and all the mathematical concepts of a set can be applied to it as a way of formalization and handling. A RO must fulfill the properties of: Query retrieval (storing every query performed), integrity update (updating structural changes in the query retrieval system), autopoietic behaviour (capacity of self creation), support for logical operators (mechanisms of set handling), and self reasoning over the query set (capacity of performing logical operations over the query system) [17]. The advantage of implementing RO relies: Speed on the query process, incremental nature, and self containment of the knowledge structure in a single file.

The RO consists of an extension which adds a new class to the base ontology with the needed schema for the reflexivity; in our implementation, we call this the "ReflexiveOntologyQueryStorer class" (fig. 4).

The extension hangs from the OWL:Thing super class and it has the following OWL properties (Table 1).

Table 1. Reflexive class properties

Property	Type	Comment
isQueryComplex	Data	Boolean
QueryDefinition	Data	String
QueryMapsToIndividuals	Object	Collection of individuals

The last part of the implementation is the reflexiveness itself and it provides the ontology (programmatically) with a mechanism to perform queries and some logic on the queries that allows the handling of the reflexiveness. For our implementation we used protégé, its API's and the OWL-DL subspecies of the OWL specification [18].

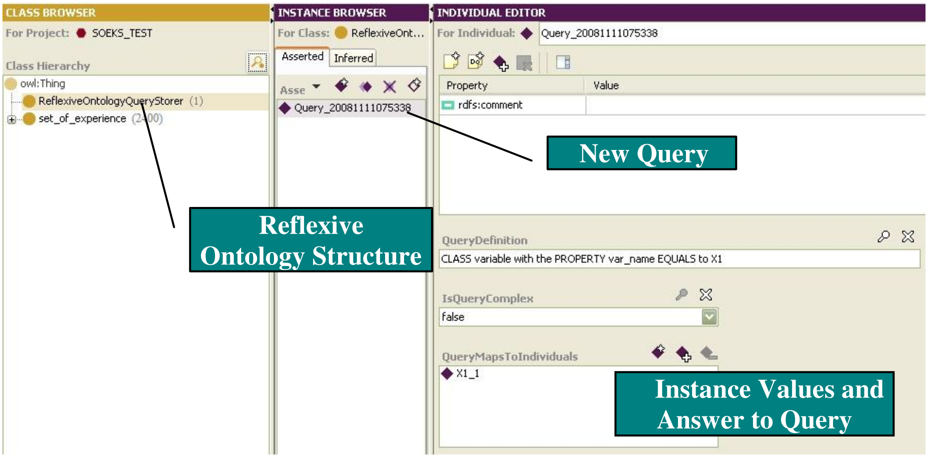
A query is used to exemplify this case study. The query is defined in the code as:

```
public static String SIMPLE_RFLEXIVE_QUERY="CLASS variable
with the PROPERTY var_name EQUALS to X1";
```

Notice that this is a value type query. Such query is written in a human-like readable form which means "retrieve all the variables of the ontology that have the variable name X1".

The execution of the code offers information about the type of query executed and the successful saving of the query executed with results within the Reflexive Ontology Structure. Following in figure 5, the results can be seen as a query successfully executed with the new instance in the SOEKS-OWL transformed into a Reflexive Ontology:

```
---------------------------START------------------------
Testing Simple query : CLASS variable with the PROPERTY
var_name EQUALS to X1
--------------------------------------------------------
... saving successful.
File modification saved with 0 errors.
```



QueryDefinition: CLASS variable with the PROPERTY var_name EQUALS to X1

IsQueryComplex: false

QueryMapsToIndividuals: X1_1

```
-------------------------------------END----------------
```

Fig. 5. Query performed on the RO geothermal chromosome

Such query performed over 2400 instances takes about 25 seconds; however, after the first time, a new similar query could take about 0.05 seconds.

4 Conclusions

This paper presents a case study for the implementation of knowledge engineering in the renewable energy field. It explains a practical case of collecting research experience from a geothermal laboratory, and then, use this information for the construction of a geothermal chromosome of an Energy Decisional DNA.

Decisional DNA and the Set of Experience were applied as a knowledge representation structure for gathering the geothermal experience. Afterwards, they were implemented in an ontology structure with reflexivity characteristics in order to use it as a tool for decision making processes that can enhance different systems with predicting capabilities and facilitates knowledge engineering processes inside decision making.

References

1. Arnold, W., Bowie, J.: Artificial Intelligence: A Personal Commonsense Journey. Prentice Hall, New Jersey (1985)
2. Asif, M., Muneer, T.: Energy supply, its demand and security issues for developed and emerging economies. Renewable Sustain Energy Rev. 111, 388–413 (2007)

3. Blakeslee, S.: Lost on Earth: Wealth of Data Found in Space. New York Times. March 20, C1 (1990)
4. Chau, K.W.: A review on the integration of artificial intelligence into coastal modelling. J. Environ. Manage. 80, 47–57 (2006)
5. Chau, K.W.: A review on integration of artificial intelligence into water quality modelling. Mar. Pollut. Bull. 52, 726–733 (2006)
6. Corti, L., Backhouse, G.: Acquiring qualitative data for secondary analysis. Forum: Qualitative Social Research 6, 2 (2005)
7. Feigenbaum, E., McCorduck, P.: The Fifth Generation. Addison-Wesley, Reading (1983)
8. Humphrey, C.: Preserving research data: A time for action. In: Preservation of electronic records: new knowledge and decision-making: postprints of a conference - symposium 2003, pp. 83–89. Canadian Conservation Institute, Ottawa (2004)
9. Johnson, P.: Who you gonna call? Technicalities 10-4, 6–8 (1990)
10. Kalogirou, S.: Artificial intelligence for the modeling and control of combustion processes: a review. Prog. Energy Combust. Sci. 29, 515–566 (2003)
11. Kalogirou, S.: Artificial Intelligence in energy and renewable energy systems. Nova Publisher, New York (2007)
12. Kyung, S.P., Soung, H.K.: Artificial intelligence approaches to determination of CNC machining parameters in manufacturing: a review. Artif. Intelligence Eng. 12, 121–134 (1998)
13. Moghtaderi, B., Doroodchi, E.: Development of a novel power cycle for geothermal energy. REDI Grant (unplublished) (2007)
14. Sanin, C., Szczerbicki, E.: Set of Experience: A Knowledge Structure for Formal Decision Events. Foundations of Control and Management Sciences 3, 95–113 (2005)
15. Sanin, C., Szczerbicki, E.: Extending set of experience knowledge structure into a transportable language extensible markup language. International Journal of Cybernetics and Systems 37(2–3), 97–117 (2006)
16. Sanin, C., Toro, C., Szczerbicki, E.: An OWL ontology of set of experience knowledge structure. Journal of Universal Computer Science 13(2), 209–223 (2007)
17. Toro, C., Sanín, C., Szczerbicki, E., Posada, J.: Reflexive Ontologies: Enhancing Ontologies with Self- Contained Queries. International Journal of Cybernetics and Systems 39(2), 171–189 (2007)
18. Zhang, Z.: Ontology query languages for the semantic Web. Master's thesis. University of Georgia, Athens (2005)

A Strategy for SPN Detection Based on Biomimetic Pattern Recognition and Knowledge-Based Features

Yan Liang[1,2], Zhongshi He[1], and Ying Liu[2]

[1] College of Computer Science, Chongiqng University, Chongqing 400044, China
{liangyan82,zshe}@cqu.edu.cn
[2] Department of Industrial and Systems Engineering,
The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong SAR, China
{Liangyan.lynn,mfyliu}@polyu.edu.hk

Abstract. Image processing techniques have proved to be effective in improving the diagnosis of lung nodules. In this paper, we present a strategy for solitary pulmonary nodules (SPN) detection using radiology knowledge-based feature extraction scheme and biomimetic pattern recognition (BPR). The proposed feature extraction scheme intends to synthesize comprehensive information of SPN according to radiology knowledge, e.g. grey level features, morphological, texture and spatial context features. Using support vector machine (SVM), Naive Bayes (NB) and BPR as the classifiers to evaluate different feature representation schemes, our experimental study shows that the proposed radiology knowledge-based features can significantly improve the classification effectiveness of SPN detection from nonnodules, in terms of accuracy and F_1 value, regardless of the classifiers used. We also note that BPR can deliver a consistent performance using our knowledge-based features, even the ratios between nonnodules and nodules are quite different in the training set.

Keywords: Feature Extraction, Solitary Pulmonary Nodules Detection, Biomimetic Pattern Recognition.

1 Introduction

Lung cancer is one of the fatal diseases widely reported [8]. Due to the development of the imaging technology, computed tomography (CT) has been commonly used for the detection of pulmonary diseases. For lung caner, one of its first key signs is the appearance of solitary pulmonary nodules (SPN) on chest radiograph or CT [12]. If the nodule can be identified in the early stage, the five year survival rate of lung cancer patients increases from 14% to 49% [4]. Detection and therapy in the early stage is the most effective approach to prevent and cure the pulmonary diseases. Although the CT technology may improve the detection of lung cancer, a large amount of CT images needs to be interpreted by radiologists. Therefore, it is desirable to develop computer-aided detection algorithm to identify the suspicious lesions automatically from CT images.

Generally, nodule detection consists of three steps: (1) preprocessing to obtain the regions of interest; (2) feature extraction; (3) classification. The last two are the most

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 672–681, 2009.

important stages in designing a system for SPN detection. Many classification techniques have been applied in nodules detection, such as rule-based, linear classifier, neural network, Support Vector Machine (SVM) and so on [14]. Yongbum et al. extracted thirteen features combined with feature rules to eliminate false positive findings in nodules detection. Their feature set included grey-level and shape features, e.g. mean, standard deviation, area, circularity and irregularity [20]. Ko et al. quantified the region of interest (ROI) with diameter, location and the change in size [7]. Mousa and Khan extracted circularity, mean and variances of the edges image and detected area as features for support vector machines (SVM) to classify lung nodules from nonnodules [10]. Boroczky et al. proposed a feature subset selection method based on genetic algorithms to choose optimal ten features out from the twenty-three 2-D and 3-D features. SVM was applied as the classifier for the nodule detection [1]. In Suzuki et al.'s report [15], pixels of subregions selected from the images were directly input into the artificial neural network (MTANN) for training. As reported by Suzuki et al, It achieved 80.3% of expert finding in overall sensitivity. However, it is time consuming to train the MTANN. Zhang et al described a 3D automated scheme using cellular neural networks to detect the pulmonary nodules. After collecting information in a neighborhood and employing the local shape property, a 3D discrete-time cellular neural network (DTCNN) was implemented to give a voxel classification. Lin et al. proposed a neural network-based fuzzy model to detect lung nodules using area, mean brightness and circularity measurement as features [8].

In this paper, we propose a different feature extraction scheme based on the domain knowledge to obtain comprehensive information of the ROI. Furthermore, joint with biomimetic pattern recognition, our strategy aims to improve SPN detection. After we obtain the ROIs, the proposed feature extraction scheme is carried out to obtain the information of the suspicious lesions on the chest CT. When analyzing the related symptom of SPN according to the radiology knowledge, our feature extraction scheme is initiated from the following aspects, including morphological features, grey level, texture and spatial context features. In morphological features, important syptomes, e.g. lobulation, burr and boundary's degree of clarity, are quantified using edge detection and D-P (Douglas-Peucker) algorithm [3]. The position information of the ROI in the local CT slice and the similarity of the related region in the adjacent slices are proposed to describe the spatial context features. Next, the strategy based on the extracted features using BPR is introduced to assign ROIs into either SPN class or non-noduel class. Finally, the system outputs the detection results.

The rest of this paper is organized as follows. Section 2 gives the details of our radiology knowledge-based feature extraction scheme for SPN detection. The experiment setup is explained in Section 3. Section 4 reports our experimental results and analysis, followed by a conclusion in Section 5.

2 Feature Extraction Schemes

A SPN is currently defined as a single spherical lesion, not more than 3 cm in diameter, completely surrounded by lung cells without any associated atelectasis or lymphadenopathy [16]. Feature extraction is one of the most important steps in automatic SPN detection. When examining the obtained ROI after segmentation, the suspicious regions may turn out to be blood vessels, bronchi and nodules. As a matter of fact, the suspicious lesions have their own medical signs. To extract powerful features, we propose some useful approach to characterize nodules based on the radiology knowledge. Basically, the nodules medical symptoms are divided into the following four classes: edge characteristic (e.g. lobulation, spiculate protuberance and blur), shape characteristic (e.g. circularity and size), density internal characteristic (e.g. calcification and cavitation), and location (e.g. lobe position). Accordingly, we design our feature extraction scheme from the following aspects: morphological and shape, grey level and texture, and spatial context features, in order to produce more objective descriptions.

2.1 Morphological Feature

Morphological characteristics can be easily identified in CT images and they are very helpful in suggesting pulmonary diseases. We introduce several approaches to extract relevant information for morphological features based on the radiology knowledge.

Cavitation and Shape Characteristics. Cavitation is one important medical signs.

Radiology knowledge 1: Cavitation is necrotic tissue within nodules, and appears as low density region in CT images [16].

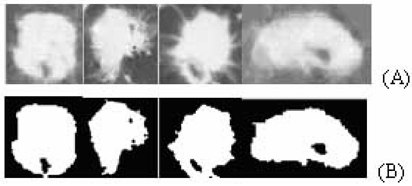


Fig. 1. Process of cavitations: (A) nodule; (B) binary image of (A)

In order to encode the radiology knowledge, the ROI (Figure 1 (A)) is transformed into a binary image (Figure 1 (B)) by OTSU [11]. Then the following features f_1-f_3 are calculated to represent the cavitation: the number of cavitation regions (dark regions within the ROI in Figure 1 (B)), the area of cavitation region, and the area ratio of cavitation region to ROI region.

Radiology knowledge 2: The shape of nodules is generally spherical on the cross slices, shown as a circle in the single slice. On the contrary, the shape of blood vessels is generally oblong as they are parallel to the slices [9].

Shape features, such as perimeter, area and roundness of the ROI, are commonly used as basic medical measurements. Thus, we firstly focus on their shape features based on the domain knowledge. The shape features include f_4-f_9: perimeter, area, circularity, elongation, slightness and variance of area [6].

Lobulation and Spiculated Characteristics. The border of nodules can be smooth, lobulated(consisting multiple lobes) or spiculated (with irregular border). The first one

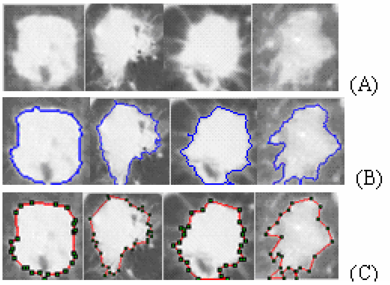


Fig. 2. Process of contour vertices: (A) nodule; (B) contour in blue; (C) key vertices in red

points toward benignity and the latter two give suggestions about malignancy. We introduce a different approach to represent the lobulation and spiculated features.

Firstly, the ROI is transformed into a binary image using OTSU [11] (Figure 2 (A)). Secondly, the contour sequence is lined out and then stored in an array (Figure 2 (B)). Thirdly, the contour vertices are marked out using Douglas-Peucker algorithm [3], which is a popular approach for describing the vertices in object boundary (Figure 2 (C)). Lastly we calculate the following features as edge characteristics f_{10}-f_{14}: irregularity, minimum concave, minimum salient, concaveness and salience.

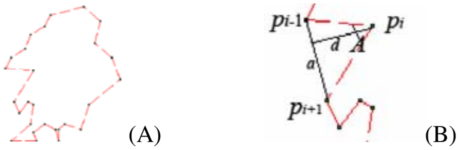


Fig. 3. Vertices of contour: (A) vertices of a nodule; (B) example of a boundary angle

The irregularity feature (f_{10}) is defined as the number of contour's vertices which indicates edge irregularity to some extent. Our experiment suggests that more vertices will appear in an irregular boundary than in a regular one (Figure 3 (A)). Given a vertex *i*, its corresponding angle can be calculated by three consecutive vertices *i*-1, *i* and *i*+1. The minimum concave feature (f_{11}) is defined as the minimum corresponding angle of the concave vertices, which is likely to be associated with a lobulation point. The minimum salient feature (f_{12}) is defined as the minimum corresponding angle of salience contour vertices, which is likely to be associated with a blur point. The concaveness feature (f_{13}) is defined as the maximum ratio of arc to chord among concave contour vertices which indicates the concaveness of the contour. We define the ratio of arc to chord as Rac_i (Figure 3 (B)):

$$Rac_i = CF_i \times \frac{d_i}{a_i} \times CD \tag{1}$$

Here CF_i is a sign to represent the specific vertex *i*, either positive or negative (+1 for salience, -1 for concave). *CD* is the boundary's degree of clarity. Rac_i is used to describe the ROI's edge characteristic (e.g. blur and lobulation). Then f_{13} is defined as f_{13}

= *max*{-Rac_i | Rac_i < 0, *i* = 1,2,3...*n* }. The salience feature (f_{14}) is defined as the maximum ratio of arc to chord among salience contour vertices, defined as: f_{14} = *max*{ Rac_i | Rac_i >0, *i* = 1,2,3...*n* }.

Boundary's Degree of Clarity. The boundary's degree of clarity is a contrast factor between lung parenchyma and nodule region. Our preliminary experiment indicates that if the nodule boundary is difficult to be distinguished in CT images, then the contour lined out using OTSU segmentation will be deformed. Here, we propose an approach to extract the features of contour clarity.

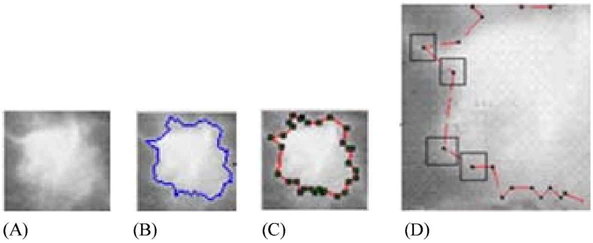


Fig. 4. Process of boundary's degree of clarity factor extraction: (A) nodule; (B) contour in blue; (C) vertices of contour in black; (D) 11×11 region with vertices as centers

Firstly, we use the same approach in 3.2.2 to obtain the sequence of contour vertices. Secondly, given each vertex *i* as a center, we calculate the grey-level mean and variance of its 11×11 sized region, shown in Figure 4. Thirdly, we compute the average mean and variance for feature f_{15}-f_{16}. In the experiment, the average variance of the contour vertices regions is proportional to the boundary's degree of clarity. Hence, the boundary's degree of clarity is defined as the average variance of the contour vertices regions and marked as *CD*. According to the medical knowledge, *CD* is also used as a parameter in calculating other features related to boundary clarity features, such as f_{13}-f_{14} concaveness and salience.

2.2 Grey Level Feature and Texture Features

Grey level features are the basic elements in describing ROIs, which are associated with density medical signs. The following 3 grey level features are used to describe the density medical sings: mean, variance and histogram energy of ROIs (f_{17} – f_{19}).

The features above can be easily observed. We try to extract the inner texture features existed in the detected structures as well as in the neighborhood of ROI concerned. Then, ten texture features f_{20}-f_{29} are derived based on co-occurrence matrices.

2.3 Spatial Context Features

The ROI's position in a single slice and its spatial structure suggest the probability of SPN occurrence.

Radiology knowledge 3: Anatomically, bronchi and blood vessels present a tree structure which is relatively stable. The shape of nodule is generally spherical and its position varies from case to case [18].

Radiology knowledge 4: The thickness of a blood vessel becomes smaller as it approaches the lung wall; however, the lung nodule's thickness is generally larger than the ordinary thickness of blood vessels at every position on the nearby slices.

In order to describe the location and spatial features of ROI, we develop some simple features according to the medical knowledge above. Radiology knowledge 3 and 4 indicate that the distance between nodule and lung wall and the distance between nodule and hilum are typical parameters to describe the nodule's position.

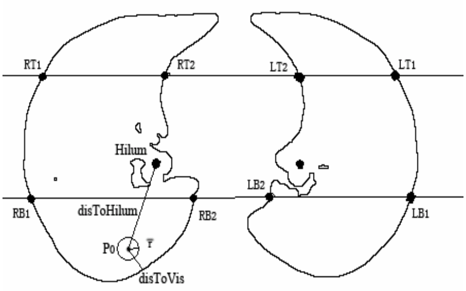


Fig. 5. Location Features of Pulmonary Nodules in Single CT Slice

Therefore, the lung region is partitioned into three parts: upper, middle and bottom to model lobes, shown in Figure 5. Next, we calculate the following features f_{30}-f_{34}: Lung side position of nodule (right or left), Lung lobe position of nodule, Distance from nodule center to lung hilum (*disToHilum*), Distance from nodule center to lung wall (*disTovisceral*) and Distance ratio of *disToHilum* over *disToVisceral.*

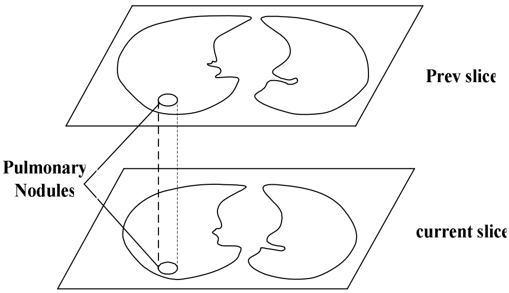


Fig. 6. Adjacent slices of pulmonary nodules

Radiology knowledge 5: The shape of lung cancer is generally spherical, and it is seen as a circle in the cross section. On the contrary, the shape of blood vessels which are parallel to the slice image is generally oblong [5].

To describe the spatial characteristics, we first identify the ROI's corresponding regions. Given the coordinates of ROI in the current slice, the corresponding region can be obtained in the adjacent slices (Figure 6). Then we describe these regions using the aforementioned features. Finally, we calculate the similarity (f_{35}) between the ROI in the current slice and its corresponding region in the previous slice and also the similarity (f_{36}) between the ROI and its corresponding region in the next slice.

3 Experiment Setup

CT images with solitary pulmonary nodules and normal cases, which have been confirmed by pathological examination, were provided by a hospital. Using the segmentation algorithm in [8], ROIs were extracted from the chest CT images including vessels, bronchus and SPNs. The dataset consists of 489 ROIs, including 234 nonnodules and 255 nodules. The ROIs were divided into four groups with similar number of nodules and nonnodules.

The ROIs were described using our proposed feature scheme mentioned in Section 2. The proposed features intend to describe the ROI from multiple aspects according to the radiology knowledge. Following our feature extraction scheme, 36 features were obtained, including 3 grey level features, 16 morphological features, 10 texture features and 7 spatial context features. Next, we build a BPR classifier for SPN identification coupled with our features. Finally, we evaluate the effectiveness of our approach, i.e. the feature extraction scheme coupled with BPR, for SPN detection.

Different from traditional pattern recognition methods which rely on statistical decision theory and emphasize optimal separation of inhomogeneous samples in feature space, e.g. linear discriminant analysis and SVM, BPR intends to imitate the recognition mechanism of a biological system [13]. The concept of BPR is that it tries to obtain an optimal subspace covering samples in each class with complicated geometrical forms in the multidimensional space. In order to recognize each class, BPR is based on the theory of high-dimensional manifold in point-set topology and uses multiple hyper-spheres to cover the sample space which contains all the samples in the same class. This technique can reasonably cover the distribution of all samples which belong to the same class in the feature space by a high-dimensional geometric complex. It also describes the topological characters of samples. Guided by the idea of "cognition", BPR has been reported to achieve the optimal covering of each sample class in feature space using high-dimensional space geometry method and multi-weight neuron theory [17].

For experiment design, due to the size of samples, the experiments were reported using four-fold cross-validation. The classifier outputs are compared with the medical diagnosis results (ground truth), where nonnodules are assigned the value of 0.0 and nodules are assigned 1.0.

In the first experiment, we adopted BPR as the classifier coupled with our radiology knowledge-based features for SPN detection. Two benchmarking algorithms, i.e. Naive Bayes (NB) and SVM, were selected to compare the performance generated based on our knowledge-based feature set (FS1) and the feature set (FS2) used in literature [8]. Secondly, we examined our approach (BPR coupled with our radiology knowledge-based features) under different training data sets where the ratios of nonnodule to nodule are 1:1.1, 1:3 and 3:1.

- Feature set 1 (FS1) consists of 36 features extracted using our proposed scheme
- Feature set 2 (FS2) contains 3 features of ROI used in [8], including area, mean brightness and circularity.

NB is simple, well understood and faster to use [2]. SVM has been widely applied in various applications, such as face recognition and text categorization [19]. We have compared the results generated in six pairs of settings between three classifiers (BPR, SVM and NB) and two feature sets (FS1 and FS2).

In terms of evaluation metrics, firstly, accuracy and F_1 are adopted. Accuracy refers to the ratio between the total number of true objects and the total number of the test objects. F_1 is an evenly weighted harmonic mean of precision and recall. Additionally, true positive rate (TPR) and false positive rate (FPR) are also reported. TPR is also called sensitivity or recall in some field. For cross-validation, the macro averages of the aforementioned metrics are computed.

4 Experiment Results

Table 1 presents the performances of three classifiers (BPR, SVM and NB) based on two different feature sets (FS1 and FS2). When our feature extracted scheme is adopted, BPR, SVM and NB have achieved an overall performance of 94.55%, 95.35% and 91.46% in F_1 respectively, about 10%, 15% and 32% higher than those using FS2. In terms of TPR, our feature set FS1 improves the performance compared to those based on FS2 (for BPR 87.07%% up to 94.92%, for SVM 82.60% up to 96.09% and for NB 49.78% down to 89.83%). We also observe that in terms of FPR, FS1 helps three classifiers to decrease the FPR to around 6%, lower than those based on FS2 (for BPR 21.34% down to 6.47%, for SVM 28.82% down to 6.04% and for NB 20.50% down to 6.87%). The results reveal that our radiology knowledge-based feature scheme helps to better distinguish SPN from nonnodules. This is largely because our feature extraction scheme has indirectly encoded the domain knowledge about SPN's characteristics into some quantitative features.

Table 1. SPN detection using BPR, SVM and NB over FS1 and FS2

(%)	Acc	F_1	TPR	FPR	(%)	Acc	F_1	TPR	FPR
FS1+BPR	94.26	94.55	94.92	6.47	FS2+BPR	83.03	84.23	87.07	21.34
FS1+SVM	95.08	95.35	96.09	6.04	FS2+SVM	77.17	78.97	82.60	28.82
FS1+NB	91.40	91.46	89.83	6.87	FS2+NB	64.02	58.51	49.78	20.50

Table 2 shows the performance of SPN detection using our proposed feature set coupled with BPR. These results are generated where nonnodule and nodule are in three different ratios (1: 3, 1: 1.1 and 3: 1) in the training set. Our first observation is that the performance of our approach using BPR based on FS1 is consistent under different ratios of nonnodule to nodule in training set.

Table 2. SPN detection using BPR based on FS1 under using different training sets with different ratios between nonnodule and nodule

Ratio of nonnodule to nodule	Acc (%)	F_1 (%)	TPR (%)	FPR (%)
1: 1.1 (234:255)	94.26	94.55	94.92	6.47
1:3 (255/3: 255)	93.75	94.01	94.40	6.90
3:1 (234: 234/3)	94.83	95.03	94.83	5.20

When training with a balanced data set (1: 1.1), our approach achieves 94.55% and 94.92% in F_1 and TPR respectively. When we switched the proportion to imbalance (1:3 and 3:1), our approach still manages to obtain comparable results with balanced training set (94.01% and 95.03% in F_1 respectively). The results suggest that our approach using BPR based on our proposed feature set is largely insensitive to ratio of nonnodule to nodule in training set.

Using the proposed features coupled with BPR, our approach offers a steady performance given different ratios of nonnodules and nodules in the training set. Moreover, our approach has generated a comparable performance compared to SVM based on FS1.

5 Conclusions

In this paper, we have presented a strategy to improve SPN detection using our proposed radiology knowledge-based feature extraction scheme and BPR, a classifier which intends to imitate the recognition mechanism of biological systems. The feature extraction scheme is initiated from morphological features, grey level, texture and spatial context features based on radiology domain knowledge. Then based on our knowledge-based features extracted, we use BPR as the classifier to assign ROIs into either SPN or nonnoduel classes. To evaluate our approach, we compare the performance given different training sets with various ratios of nonnodules and nodules. To evaluate the effectiveness of our feature scheme, we benchmark our features with the feature set used in literature [8] using BPR, SVM and NB as the classifiers to assign ROIs into either SPN or nonnoduel classes. The experiment study shows that our approach, using knowledge-based features combined with BRP, offers a consistent performance under different ratios of nonnodules and nodules in the training set. Moreover, our features help to generate better performance for SPN detection regardless of the classifiers used.

Acknowledgement

This work is partly supported by Science and Technology Development Major Program of Chongqing (CSTC, 2008AB5038) and Natural Science Foundation Project of Chongqing (CSTC, 2007BB2134).

References

1. Boroczky, L., Zhao, L., Lee, K.P.: Feature Subset Selection for Improving the Performance of False Positive Reduction in Lung Nodule CAD. IEEE Transactions on Information Technology in Biomedicine 10, 504–511 (2006)

2. Colas, F., Brazdil, P.: Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks. In: Artificial Intelligence in Theory and Practice (2006)
3. Douglas, D., Peucker, T.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer 10, 112–122 (1973)
4. Gurcan, M.N., Sahiner, B., Petrick, N., Chan, H.-P., Kazerooni, E.A., Cascade, P.N., Hadjiiski, L.: Lung nodule detection on thoracic computed tomography images: Preliminary evaluation of a computer-aided diagnosis system. Medical Physics 29, 2552–2558 (2002)
5. Kanazawa, K., Kawata, Y., Niki, N., Satoh, H., Ohmatsu, H., Kakinuma, R.: Computer-aided diagnosis for pulmonary nodules based on helical CT images. In: Fourteenth International Conference on Pattern Recognition. Proceedings (1998)
6. Kanazawa, K., Kawata, Y., Niki, N., Satoh, H., Ohmatsu, H., Kakinuma, R., Kaneko, M., Eguchi, K., Moriyama, N.: Computer-aided diagnosis for pulmonary nodules based on helical CT images. In: Nuclear Science Symposium. IEEE, Los Alamitos (1997)
7. Ko, J.P., Betke, M.: Chest CT: Automated Nodule Detection and Assessment of Change over Time–Preliminary Experience. Radiology 218, 267–273 (2001)
8. Lin, D.-T., Yan, C.-R., Chen, W.-T.: Autonomous detection of pulmonary nodules on CT images with a neural network-based fuzzy system. Computerized Medical Imaging and Graphics 29, 447–458 (2005)
9. Marten, K., Engelke, C., Seyfarth, T., Grillh, A., Obenauer, S., Rummeny, E.J.: Computer-aided detection of pulmonary nodules: influence of nodule characteristics on detection performance. Clinical Radiology 60, 196–206 (2005)
10. Mousa, W.A.H., Khan, M.A.U.: Lung nodule classification utilizing support vector machines. In: 2002 International Conference on Image Processing (2002)
11. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Sys., Man., Cyber. 9, 62–66 (1979)
12. Shah, S.K., McNitt-Gray, M.F., Rogers, S.R., Goldin, J.G., Suh, R.D., Sayre, J.W., Petkovska, I., Kim, H.J., Aberle, D.R.: Computer-aided Diagnosis of the Solitary Pulmonary Nodule1. Academic Radiology 12, 570–575 (2005)
13. Shoujue, W., Jiangliang, L.: Geometrical learning, descriptive geometry, and biomimetic pattern recognition. Neurocomputing 67, 9–28 (2005)
14. Sluimer, I., Schilham, A., Prokop, M., van Ginneken, B.: Computer analysis of computed tomography scans of the lung: a survey. IEEE Transactions on Medical Imaging 25, 385–405 (2006)
15. Suzuki, K., Armato, S.G., LI, F., Doi, K.: Massive training artificial neural network (MTANN) for reduction of false positives in computerized detection of lung nodules in low-dose computed tomography. Medica Physic. 30, 1602–1617 (2003)
16. Tang, A.W.K., Moss, H.A., Robertson, R.J.H.: The solitary pulmonary nodule. European Journal of Radiology 45, 69–77 (2003)
17. Wang, S., Huang, Y., Cao, Y.: Study on Text-Dependent Speaker Recognition Based on Biomimetic Pattern Recognition. In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3972, pp. 158–164. Springer, Heidelberg (2006)
18. Wiemker, R., Rogalla, P., Zwartkruis, A., Blaffert, T.: Computer-aided lung nodule detection on high-resolution CT data. In: Medical Imaging 2002: Image Processing (2002)
19. Xiaosheng, P., Jiangping, K., Alan Wee-Chung, L.: Lip Contour Extraction Based on Support Vector Machine Add Support. In: Congress on Image and Signal Processing (2008)
20. Yongbum, L., Hara, T., Fujita, H., Itoh, S., Ishigaki, T.: Automated detection of pulmonary nodules in helical CT images based on an improved template-matching technique. IEEE Transactions on Medical Imaging 20, 595–604 (2001)

Generating Bidirectional Links for Web Annotation Stickies

Hiroyuki Sano, Tadachika Ozono, and Toramatsu Shintani

Dept. of Computer Science and Engineering
Graduate School of Engineering, Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya, Aichi, 466-8555 Japan
{hsano,ozono,tora}@toralab.ics.nitech.ac.jp

Abstract. We propose a web annotation system which automatically generates bidirectional links between annotation stickies referencing similar information and subsequently categorizes them. Our system has the functionality of stickies to web pages. The stickies allow for important parts of a web page which contains large amounts of data to be highlighted. Such stickies and links can be used as user preferences, and have the potential to become a much better alternative to bookmarks and tags.

1 Introduction

We have implemented a web annotation system for web contents, which automatically generates bidirectional links between annotation stickies. Compared with bookmarks and tags, bidirectional links contain much more specific information. Bookmarks and tags are groups or collections of links pointing to web pages which have some significance to the user. Bookmarks and tagging systems enable users to specify only the web page of interest, while bidirectional links are sets of links which refer to specific content of significance to the user. Information obtained from contents collections can be utilized in the development of web search engines and recommendation systems based on new points of view, thus differing significantly from current search engines and recommendation systems.

Users can annotate web contents by providing comments, tags and links inside stickies. Web pages contain texts, images, and other types of information which are often related to more than one topic. By using our system, users can point out specific contents more accurately than by using bookmarks in web browsers or current tagging systems, and users also make bidirectional links between stickies.

Our system contains agents that attach bidirectional links to the stickies in order to cross-reference similar contents in the system. The agents monitors the stickies which users place and generates bidirectional links to the stickies which were placed on similar contents. As a result, users can traverse stickies referencing similar content by using the bidirectional links, which are generated between not only the stickies that the user has placed, but also between those that other users have placed. This enables users to find previously unknown web pages.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 682–690, 2009.

The rest of the paper is organized as follows. Section 2 describes how the agents generate bidirectional links between stickies. In Section 3, we propose a web annotation system, in which the agents automatically generate bidirectional links. Experiments are reported in Section 4. Section 5 review related work in the area of web annotation. Finally, we conclude the paper in Section 6.

2 Generating Bidirectional Links by Agents

The system creates bidirectional links between stickies, where the links present the relations between the contents of different stickies in a clear manner. The system comprises a web agent called a 'biLink agent' which helps users to create bidirectional links.

2.1 Bidirectional Links between Stickies

Weblogs implement a function called the 'trackback', which informs weblog authors about what kinds of web pages are linking to articles in the weblog. In this sense, the trackback feature makes weblogs bidirectional. In our system, a biLink agent generates links between stickies with a similar content based on the concept of trackback, and as a result users can traverse the stickies by using those links.

A biLink agent implements a function for automatic generation of bidirectional links between stickies. A biLink agent is used for keeping track of the stickies placed by the user. When a user places a sticky, the agent extracts the text around the content where the sticky is placed by looking at the Document Object Model tree. This process is based on heuristics, in other words, on the text around image files or flash files which describes those files[1]. The extracted text is sent to an agent on a web server, which analyzes the text and classifies the sticky in accordance with a classification method explained in Section 2.3. Subsequently, the agent automatically generates bidirectional links between stickies placed on similar contents.

2.2 biLink Agent

Figure 1 illustrates the principle behind the biLink agent. The figure contains six stickies, (1) to (6), which the user has placed using the system. The (A) indicates a universal set of stickies, and the biLink agent classifies those stickies. In Figure 1, the arrows indicate the process of classifying stickies. Let those six stickies be classified into three sets, (a), (b), and (c), as follows.

- (a)={(2), (4)}
- (b)={(1), (5), (6)}
- (c)={(3)}

The biLink agent generates bidirectional links between stickies on the basis of the results of the classification. The set (a) contains two stickies, (2) and (4),

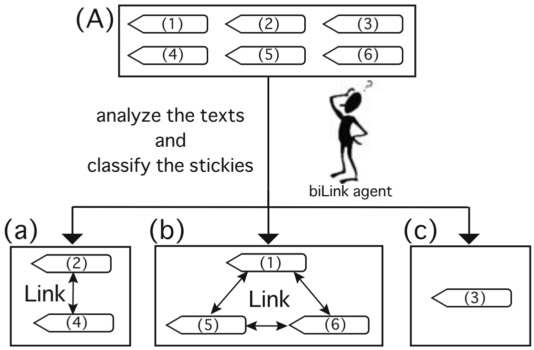


Fig. 1. Example of the operating principle of the biLink agent

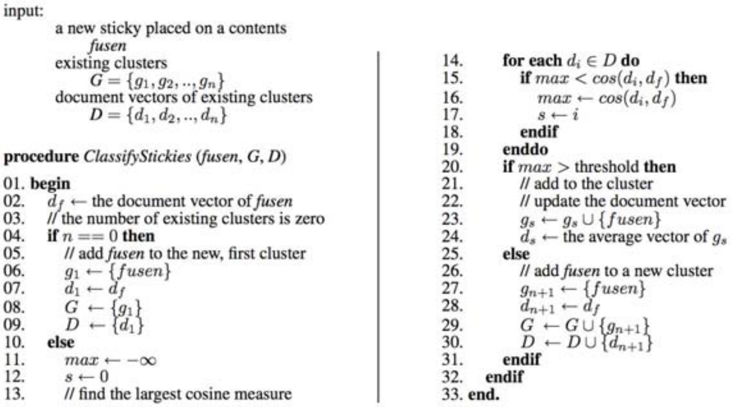
and therefore the biLink agent generates bidirectional links between (2) and (4). The set (b) contains three stickies, (1), (5), and (6), and as a result the biLink agent generates bidirectional links between each pair of stickies (1), (5), and (6). As the set (c) contains only one sticky, (3), the biLink agent does not need to generate any links.

2.3 Classification Method of Stickies

Yang, et al. examined some approaches to classify hypertext documents[2]. We present a method which the biLink agent uses to classify the stickies. The biLink agent uses MeCab[1] to parse a web page containing the information on which a user has placed a sticky and decides the index terms of the web page. The biLink agent then calculates the evaluations of the index terms by the Term Frequency and the Inverse Document Frequency(TF-IDF). TF is the number of times the index term appears in the web page containing the information which the user has referenced with a sticky. Since the system is a web-based application, the biLink agent uses the total index size of the Yahoo! API as the number of all documents, and uses the number of results which the Yahoo! API could obtains as the number of documents where the index term appears, when the agent calculates the IDF.

The similarity between documents in classifying stickies is calculated by using a cosine measure based on the Vector Space Model. Each dimension of a document vector corresponds to a separate term, and each component corresponds to an evaluation of the term. However, the biLink agent performs the calculation by assigning a certain weight to the content which is referenced with a sticky. The term ‘content’ here indicates the nearest block-level element based on Document Object Model, where the tracing is in the direction from the node where the sticky is placed toward parent nodes.

[1] http://mecab.sourceforge.net/



```
input:
    a new sticky placed on a contents
        fusen
    existing clusters
        $G = \{g_1, g_2, .., g_n\}$
    document vectors of existing clusters
        $D = \{d_1, d_2, .., d_n\}$

procedure ClassifyStickies (fusen, G, D)

begin
    $d_f \leftarrow$ the document vector of fusen
    // the number of existing clusters is zero
    if $n == 0$ then
        // add fusen to the new, first cluster
        $g_1 \leftarrow \{fusen\}$
        $d_1 \leftarrow d_f$
        $G \leftarrow \{g_1\}$
        $D \leftarrow \{d_1\}$
    else
        $max \leftarrow -\infty$
        $s \leftarrow 0$
        // find the largest cosine measure
        for each $d_i \in D$ do
            if $max < cos(d_i, d_f)$ then
                $max \leftarrow cos(d_i, d_f)$
                $s \leftarrow i$
            endif
        enddo
        if $max >$ threshold then
            // add to the cluster
            // update the document vector
            $g_s \leftarrow g_s \cup \{fusen\}$
            $d_s \leftarrow$ the average vector of $g_s$
        else
            // add fusen to a new cluster
            $g_{n+1} \leftarrow \{fusen\}$
            $d_{n+1} \leftarrow d_f$
            $G \leftarrow G \cup \{g_{n+1}\}$
            $D \leftarrow D \cup \{d_{n+1}\}$
        endif
    endif
end.
```

Fig. 2. The procedure for classification

Document vectors in the system are calculated by the following formula.

$$d_i = (w_{i1}, w_{i2}, \cdots, w_{iM})^T + \alpha(v_{i1}, v_{i2}, \cdots, v_{iM})^T$$

w_{ij} indicates an evaluation of term $t_j (j = 1, 2, \cdots, M)$ in a document number i, v_{ij} indicates an evaluation of term t_j in a content of a document number i, M indicates the number of different terms in a unit of documents, and α is the weight. Thus, by using document vectors calculated by assigning a certain weight to the content referenced with a sticky, the system can classify web pages containing multiple topics with a high degree of accuracy.

Next, the cosine measure is calculated by the following formula using two document vectors, d_1 and d_2.

$$cos(d_1, d_2) = \frac{d_1 \cdot d_2}{||d_1|| \cdot ||d_2||}$$

Here, the lower the degree of the two vectors, the larger the cosine measure. When the system classifies stickies, a new cluster is created in which the average vector of document vectors present in the cluster is taken as the cluster vector.

Figure 2 outlines the procedure used by the biLink agent to classify stickies. The biLink agent analyzes a content on which a user has placed the first sticky, calculates the document vector, and generates the first cluster which contains only that first sticky. After that, as more stickies are placed on parts of the page with different contents, the biLink agent analyzes the content in those parts, and calculates the document vectors as well as the similarity between the document vectors and the document vectors of the clusters which already exist. If the similarity is greater than a predefined threshold, the biLink agent adds the sticky to the cluster and updates the document vector of the cluster with the average vector. If the similarity is lower than the threshold, the biLink agent generates a new cluster and adds the sticky to the new cluster.

The weight α and the similarity threshold are decided on the basis of the results performed by a person. A certain number of web pages are collected at random and classified manually, after which the system also classifies them. In order to match the results of the manual classification with the results of the classification performed by the system, we adjusted α and the threshold. Eventually, the most satisfactory level of conformance was attained when α was 25 and the threshold was 0.15.

When a user starts using the system, there are no stickies placed on the page, and therefore no clusters. After the first sticky is placed on the page, the first cluster is generated. After that, as the user adds more stickies, the biLink agent classifies those stickies. If a sticky is classified in to an existent cluster, the biLink agent generates bidirectional links between the sticky and all other stickies in the same cluster, and the final result is that the bidirectional links form a complete graph. As the href attribute in each anchor element only specifies one source, the system shows all links, as in (4) in Figure 4.

3 Web Annotation Stickies with Bidirectional Links

Our web annotation system enables users to place stickies on web contents and to provide comments in relation to the content referenced by the stickies. We refer to the activity of placing stickies and writing comments as 'Fusen Annotation'. Fusen means 'paper annotation sticky' in Japanese. Users can place annotation stickies on all types of contents on web pages, including text data, images, and so on, by using the web annotation system.

3.1 Web Annotation System

In the system, a web agent, which is referred to as a 'biLink agent', keeps track of the stickies which users have placed on web pages. The biLink agent is constructed from a page agent and a base agent, using the web agent model 'MiSpider'[3]. The page agent sends to the base agent the web content on which a user has placed a sticky. Then the base agent classifies the stickies by using information which it has received from the page agent, and generates bidirectional links between the stickies placed on similar contents.

Figure 3 shows the outline of the system. The system comprises a Fusen client and a Fusen server. The Fusen client, which is written in JavaScript, runs on the user's web browser and acts as a page agent. The Fusen client provides an interface for placing stickies on web contents. The Fusen server is the system which saves the stickies which users have placed. The Fusen server is a proxy server, which acts as a base agent, and a database for saving the properties of the stickies.

When a user requests a web page from the web annotation system, web browser obtains the HTML source code of the web page from the proxy server, which contains the database with the properties of the stickies. The proxy server accesses the database and sends an inquiry regarding whether the HTML source

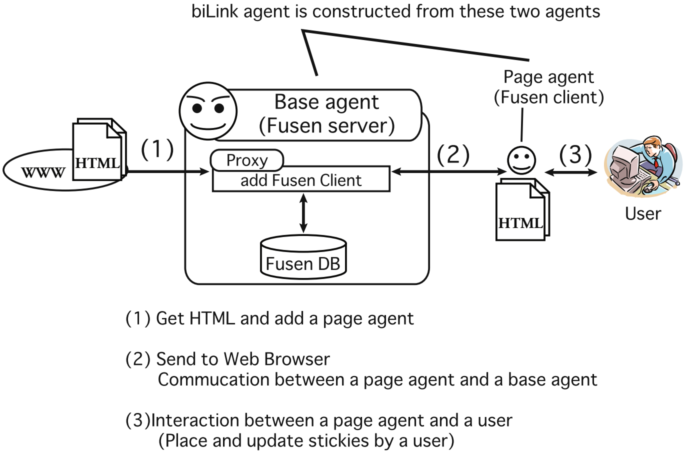

biLink agent is constructed from these two agents
Base agent
(Fusen server)
Page agent
(Fusen client)
WWW
HTML
(1)
Proxy
add Fusen Client
(2)
(3)
User
HTML
Fusen DB
(1) Get HTML and add a page agent
(2) Send to Web Browser
Commucation between a page agent and a base agent
(3)Interaction between a page agent and a user
(Place and update stickies by a user)

Fig. 3. Web Annotation System

code of the web page has been saved to the database. If the page has been saved, the proxy server sends the HTML code to the web browser. If not, the proxy server obtains the code from the relevant web server, adds a Fusen client to the HTML code, and sends the HTML code thus generated to the web browser. There are two methods for adding web services to an existing web page: one involves a proxy server[4], and the other is based on bookmarklets[5]. We adopted the proxy server method for the system. If a user places a new sticky or updates an existing sticky, the Fusen client updates the database on the Fusen server when the user leaves the web page.

3.2 Example of Execution

Figure 4 shows a screenshot of a web page on which a user has used the system to place stickies. The browser is Safari, which has been developed by Apple, and the stickies were placed on two articles in Google News.

Users can place annotation stickies on web contents as shown in Figure 4. When a user double-clicks an annotation sticky, they can see detailed information about the sticky in a dedicated popup window, which can be closed by double-clicking on the sticky again. In the example, the user has placed two annotation stickies on the page and has opened the popup window of the lower sticky. Regarding points (1) to (5), which are shown in the popup window in Figure 2, (1) indicates the date when the user placed the sticky, (2) is used when the user wishes to change the color of the sticky, (3) is a comment to the referenced content (this comment is also displayed on the sticky image), (4) shows links to similar contents (when a user clicks on a link, they can see the stickies which have been placed on similar contents), and (5) is used for deleting the sticky.

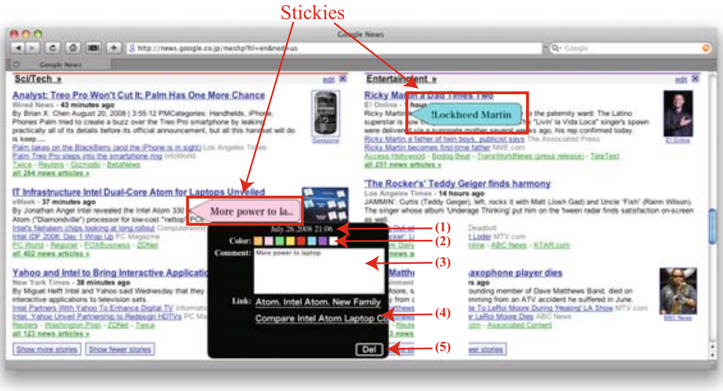


Fig. 4. An execution example of Fusen Annotation

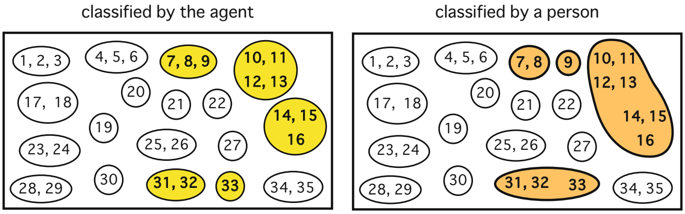


Fig. 5. Results of the classification by the biLink agent (left) and a person (right)

4 Experimental Result

We evaluated bidirectional links generated by the agents. We placed 35 stickies on different parts of web pages with multiple contents categories at random, and compared the results of the manual classification and that performed by the agents.

Figure 5 shows the results of the classification. The left image in Figure 5 presents the results of the classification performed by the agents, and the image on the right corresponds to the manual classification. Each number (1 to 35) in Figure 5 represents the number of a sticky. Stickies which are located inside the same circle have been classified into the same cluster.

The results of the classification performed by the agents are remarkably similar to those of the manual classification. The agents classify the 35 stickies into 18 clusters, and 13 clusters are exactly the same as the clusters classified by a person, which amounts to an accuracy of 72.2%. In addition, the 5 clusters which differ in their classification still closely resemble the clusters classified by a person. Thus, bidirectional links generated by the agents generally represent relations between web contents in person's perspectives.

5 Related Work

There are several annotation systems for web pages. Annotea[6][7], which was developed by W3C, is a framework which allows annotations to be placed on web pages. Annotea is a Semantic Web based project for which the inspiration came from users' collaboration problems in the Web. It examined what users did naturally and selected familiar metaphors for supporting better collaboration. Annotea is available for Firefox through the Annotea Ubimarks extension, as well as for Amaya, which is an open-source web browser developed by W3C. However, although Annotea enables users to create links from one web page to other web pages, this needs to be done manually.

There also exist social tagging systems, where users add tags to web contents and share those tags with other users[8]. Chirita, et al. propose P-TAG, a method which automatically generates personalized tags for Web pages. Upon browsing a Web page, P-TAG produces keywords relevant both to its textual content, but also to the data residing on the surfer's Desktop, thus expressing a personalized viewpoint[9]. However, to express relations between tags is difficult, although tags are useful for specifying both web pages and web contents.

Some studies show that annotations can be useful for education systems. For example, D. Giordano and S. Mineo propose a graphical annotation system for a distributed e-learning architecture. The role of annotations for e-learning is discussed and technologies for web annotation are reviewed[10]. Farzan and Brusilovsky presents their attempt to integrate annotation and adaptive navigation support for open corpusweb educational resources into a single value-added service. The AnnotatEd system offers both annotation (through highlighting and free-text comments) and adaptive navigation support (through social navigation). It works in the traditional intermediary way, by standing between a web page and its user[11].

6 Summary and Conclusions

Bidirectional links between annotations can enable users to find new pages that are difficult to find by existing search engines. Our web agent in the system automatically generates bidirectional links between annotation stickies in order to show the relations between the stickies which reference similar contents. Users can traverse the referenced content in a two-way fashion by using the links, thus coming across previously unknown web pages. Furthermore, bidirectional links are useful in that they can contain more information than social tags. We have implemented a system which allows the placement of web annotation stickies on web pages.

Generating links to stickies which have been placed by other users can be used for collecting information from the web. Moreover, by sharing the stickies between many users, a search algorithm which is based on criteria different from existing web search engines can be made available to the public. For example, users can search for information on the basis of the arrival probability in bidirectional links.

Searching with stickies enables users to find important parts of web pages instead of only the web page itself, thus increasing the efficiency of the search. By searching relevant contents cross-referenced with bidirectional links instead of all web pages, users can obtain highly precise search results. In addition, if high recall is needed, users can perform combined searches including both stickies and conventional search engines. Stickies can display useful contents more correctly than bookmarks, and therefore they are effective for high-precision information retrieval and information referencing.

References

1. Chen, Z., Wenyin, L., Zhang, F., Li, M., Zhang, H.: Web mining for web image retrieval. Journal of the American Society for Information Science and Technology 52, 831–839 (2001)
2. Yang, Y., Slattery, S.A.: A study of approaches to hypertext categorization. Journal of Intelligent Information Systems 18, 219–241 (2002)
3. Fukagaya, Y., Ozono, T., Ito, T., Shintani, T.: Mispider: a continuous agent on web pages. In: WWW 2005: Special interest tracks and posters of the 14th international conference on World Wide Web, pp. 1008–1009. ACM Press, New York (2005)
4. Sakamoto, S., Kita, H., Takase, H., Hayashi, T.: Platform for web services using proxy server. The Special Interest Group Notes of IPSJ 2002(31), 7–12 (2002)
5. Tanabe, M., Ozono, T., Itoh, T., Shintani, T.: Web servise addition system by bookmarklet using user browsing domain. In: Proceedings of the 68th National Convention of IPSJ (2006)
6. Koivunen, M.R.: Annotea and semantic web supported collaboration. Invited talk at Workshop on User Aspects of the Semantic Web (User-SWeb) at European Semantic Web Conference (2005)
7. Kahan, J., Koivunen, M.R.: Annotea: an open rdf infrastructure for shared web annotations. In: WWW 2001: Proceedings of the 10th international conference on World Wide Web, pp. 623–632. ACM Press, New York (2001)
8. Tennis, J.: Social tagging and the next steps for indexing. In: Proceedings of the 17th SIG/CR Classification Research Workshop (2006)
9. Chirita, P.A., Costache, S., Nejdl, W., Handschuh, S.: P-tag: large scale automatic generation of personalized annotation tags for the web. In: WWW 2007: Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada, pp. 845–854. ACM Press, New York (2007)
10. Giordano, D., Mineo, S.: A graphical annotation platform for web-based e-learning. In: Proc. of Multimedia and Information and Communication Technologies in Education, pp. 1255–1260 (2005)
11. Farzan, R., Brusilovsky, P.: Annotated: A social navigation and annotation service for web-based educational resources. New Rev. Hypermedia Multimedia 14(1), 3–32 (2008)

Taguchi Analysis of Milling Wear Automatic Monitoring System Based on Machine Vision Technique

Yu-Teng Liang[1,2] and Yih-Chih Chiou[1]

[1] Institute of Engineering Science, Chung-Hua University,
No. 707, Sec.2, WuFu Rd., Hsinchu, 300, Taiwan
chiou@chu.edu.tw

[2] Department of Automation Engineering, Ta Hwa Institute of Technology,
No.1, Da-Hua Rd., Qiong-Lin, Hsinchu 30740, Taiwan
lyd337@thit.edu.tw

Abstract. This study proposes a tool wear monitoring system based on machine vision technique. The tool wear of single edge rhombus micro-end-mills with mill parameters (Side Clearance Angle, different coating layer, feed rate and spindle speed) in milling 6061 aluminum alloy was experimentally investigated in this study. A $L_9(3^4)$ orthogonal array, analysis of variance (ANOVA) and signal-to-noise (S/N) were determined to know the level of importance of the machining parameters. Using Taguchi method for design of a robust experiment, the interactions among factors are also investigated. The experimental results indicate that Side Clearance Angle and coating layer are recognized to make the most significant contribution to the overall performance. The correlation was obtained by multi-variable nonlinear regression and compared with the experimental results. The experimental results showed that Using TiCN-coated micro-end-mill and setting side clearance angle at 12 degrees, spindle speed at 6000 rpm and feed rate at 0.0125 mm/rev minimized the wear on micro-end mills and maximized tool life. The confirmation tests demonstrated a feasible and an effective method for the evaluation of tool wear in milling of 6061 aluminum alloy.

Keywords: Machine Vision, Rhombus Micro-end-mill, Different Coating, Taguchi Method, Analysis of Variance.

1 Introduction

Machine vision technology [1-3] improves productivity and quality management and provides a competitive advantage to industries that employ this technology. Machining operations like drilling, milling and turning are involved in the production of most of the goods that are used everyday. In machining processes, the microminiaturization and complicated form of products has already become a worldwide trend. For this reason, the micro-end-milling process is one of the most widely used material removal processes in the microminiaturization industry. In addition, multilayer coatings were extensively used in milling tools because of their relatively high wear-resistant, hardness, anti-corrosion, and low coefficient of friction. In recent years, PVD-coated

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 691–700, 2009.

high speed steel tools [4, 5] and the micro-end-milling process [6-8] have received increased attention. There have been numerous articles published in the literature reporting the benefits of putting a thin layer of Ti-based coating onto cutting tools for improving tool life, productivity and workpiece quality [9]. On the other hand, to protect the environment and reduce production costs of 16% to 20% [10], several European countries encourage the metal cutting industry to use dry machining. This is because dry machining does not use cutting fluids. However, this drastically affects tool wear and tool life. Dry machining has two major impacts on the ecology and economy. For dry milling applications, the cutting tools can be designed in three different ways: using new tool materials, adapting new tool geometries, and applying different coating materials [11].

Traditionally, visual inspection and quality control are performed by human experts. However, human experts are difficult to find or maintain in an on-line production. For this reason, Most of the on-line identification techniques of tool wear based on the measurement of cutting forces and power signals [12-16]. These techniques can be broadly classified as direct and indirect methods based on the type of sensor measurements. In order to automated inspect tool wear in micro-end-milling processes, a cost-effective method using a machine vision, extract vertex and Statically Process Control (SPC) methods on the tool wear is proposed.

The object of this research was to study the dry milling characteristics of carbide micro-end-mills. The experiment was to milling 6061 aluminum alloy using micro-end-mills. To observe the effects of machining parameters on tool wear and tool life of the micro-end-mills, a series of dry milling tests were performed. In the experiments, two cutting parameters were examined, including feed rate and spindle speed. Moreover, the effect of side clearance angle and coating layer of cutters on tool wear was also noted. Therefore, the purpose of this research is to monitoring the tool wear of different coated milling based on the Machine vision technology of cutters tool in the milling operation. The tool wear images of the cutters are captured and processed using a machine vision system incorporating with the subpixel edge detection technique, vertex detection algorithm and SPC Inspection technique.

2 Experimental Setup and Method

Figure 1 shows the set-up of the experimental hardware. The milling tests were conducted on a LEADWELL V30 vertical machining center. The workpiece is a 6061 aluminum alloy. The experiment used single edge rhombus micro-end-mills of various side clearance angles to mill the workpiece. Two types of tungsten carbide (WC) micro-end-mills were used in this study. One has different PVD (physical vapor deposition) coating layers. The other has different side clearance angles. Each micro-end-mill is 6 mm in length and 2 mm in diameter and has a front clearance angle of 12 degrees. The geometry of the micro-end-mills is designed by ourselves and defined by the side clearance angle. Three types of PVD-coated micro-end-mills were used in the experiment. The coating materials include TiN, TiCN and TiAlN.

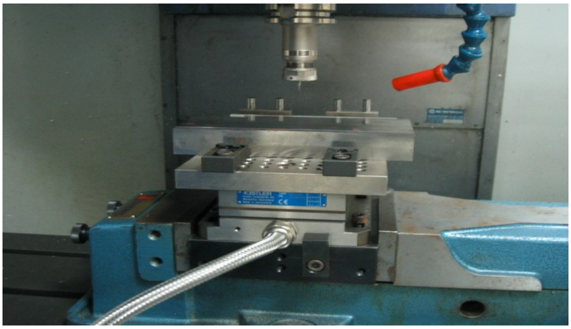


Fig. 1. Experimental setup

2.1 Taguchi Method

The Taguchi method [17] is a cost-effective and time-saving method to explore interrelationships among various parameters. Recently, it has been extensively used to determine optimum parameters. In this study, we also applied the Taguchi method to design the milling experiment to rapidly obtain an optimum combination of cutting parameters. To examine the influences of various control factors on tool wear, a $L_9(3^4)$ orthogonal array as shown in Table 1 was employed. As shown, four control factors were explored and each control factor has three levels to be examined. Here, both cutting-edge wear and side flank wear are expected to be as small as possible. Therefore, the generic form of Signal to Noise (S/N) ratio is:

$$S/N = -10 \times \log_{10}(\frac{1}{n}\sum_{i=1}^{n} y_i^2) \tag{1}$$

where y_i denotes the i^{th} measured value of wear and n denotes the total number of values to be measured. Since two kinds of wears (cutting-edge wear and side flank wear) were measured in each cutting test, n equals 2.

Table 1. Design Parameters and Levels in Milling Process

Sample	Control Factor	Level 1	Level 2	Level 3
A	Side Clearance Angle	10°	12°	14°
B	Coating Layer	TiN	TiCN	TiA1N
C	Feed Rate [mm/rev]	0.0075	0.0100	0.0125
D	Spindle Speed [rpm]	6000	6500	7000

2.2 Tool Wear Measurement System

The wear tests were conducted according to the orthogonal array matrix. The objective of this test is to find the ideal values for the two machining parameters, that is, feed rate and spindle speed and the two parameters of micro-end-mills, that is, side clearance angle and coating material. After the pre-set milling volume had been reached (0.6×10^5 mm^3), the side flank wear and cutting-edge wear were measured in an on-line machine vision system. The machine vision system for tool wear monitoring system consists of a CCD camera, a macro lens, a ring light, a PC for image processing, and suitable software. The image (640x480 pixels) of cutting edge was captured by a CCD camera via a ring-lighted. Then, the image was registered using self-developed software in order to find out an optimal tools image. Fig. 2 shows the tool wear measurement system.

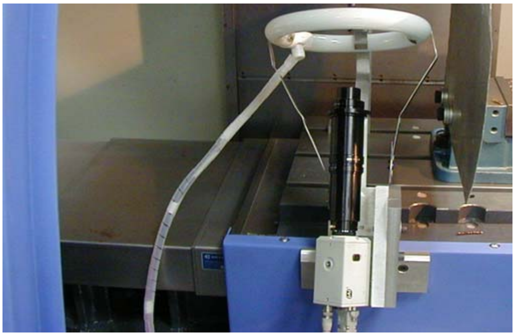


Fig. 2. Tool wear measurement system

2.3 Tool Image Processing

To automate vertex detection process, a threshold value should be selected automatically. In this research, five automatic thresholding methods have been investigated, including P-Tile method, Mode method, Iterative threshold selection method, Otsu method, and Moment preserving method. After testing with various images, the results suggest that Otsu method is a better choice when the area occupied by objects larger than one half of the total area. Otherwise, Iterative and Mode methods should be chosen. Consequently, the threshold method is automatically selected based on the actual areas occupied by the objects.ïThe flowchart for the proposed algorithm is depicted in Fig. 3.

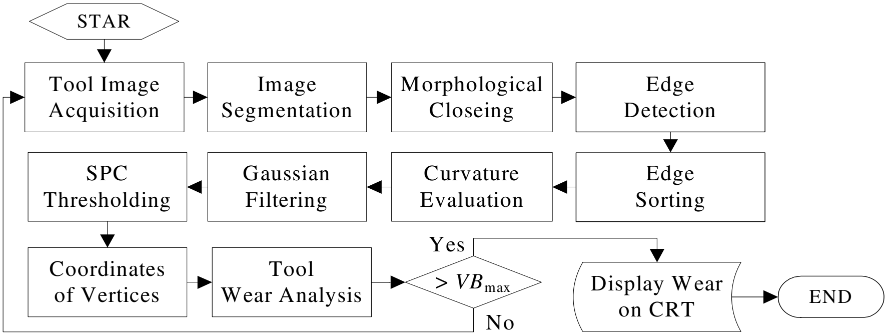


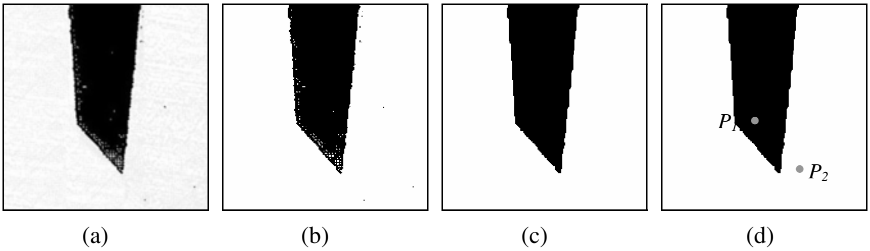
Fig. 3. Flowchart of the proposed algorithm

2.4 Tool Wear Analysis

The morphology processing was used in the image post-processing in order to obtain the optimal flank wear image. The wear of cutting edge after the image post-processing is analyzed via the aid of personal computer. The MIL (Matrox Imaging Library) software with user-friendly window interface is used to measure the crater wear width. The maximum tool wear ($VB_{\max}$) is definition by:

$$VB_{\max} = p_1 - p_2 \quad = p_1 - \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2}$$

Fig. 4 shows the images of tool segmentation before and after the image processing. The original images of micro-end-mills are shown in Fig. 4(a), Fig. 4 (b) and (c) show the binarized image and morphological operator image, respectively. Fig. 4(d) shows the detected vertices by referring to the smoothed curvature curve. Thus, after smoothed by Gaussian filter, the vertex detection method performs perfectly.



(a) (b) (c) (d)

Fig. 4. Result of the tool wear analysis

3 Experimental Results and Discussion

3.1 Tool Wear Analysis

The experimental results obtained according to the Taguchi method are tabulated in Tables 2 and 3. Table 2 shows the measured wear, both cutting-edge wear and side

Table 2. The $L_9(3^4)$ Orthogonal Table, the Measured Wears, and the S/N Ratio

No	A	B	C	D	Cutting Edge Wear (mm)	Side Flank Wear(mm)	S/N(dB)
1	1	1	1	1	0.195	0.053	16.90
2	1	2	2	2	0.093	0.114	19.66
3	1	3	3	3	0.152	0.123	17.19
4	2	1	2	3	0.106	0.193	16.15
5	2	2	3	1	0.069	0.081	22.47
6	2	3	1	2	0.074	0.252	14.62
7	3	1	3	2	0.156	0.232	14.08
8	3	2	1	3	0.220	0.206	13.42
9	3	3	2	1	0.062	0.286	13.68

flank wear, and the calculated S/N ratios. By example, the S/N ratio for test 1 can be determined as

$$S/N = -10\times\log_{10}\left(\frac{1}{2}(0.195^2+0.053^2)\right) = 16.90$$

Table 2 shows the response table and response graph derived from the nine S/N ratios as shown in Table 3. For instance, the response value for level 1 of control factor A (A_1) can be determined as

$$A_1 = (16.90+19.66+17.19)/3 = 17.914$$

As can be seen from Table 3, the preference value for the side clearance angle is 10^o, and 12^o is better than 14^o. As regards the coating layer, TiCN coated mills generate less wear than TiN coated mills, and TiN coated mills generate less wear than TiAlN coated mills. This is because TiCN coated mills have relatively high micro hardness and lower friction against dry steel. As to the feed rate, among the three values investigated, 0.0125 mm per revolution is the best choice, followed by 0.0100 mm/rev and 0.0075 mm/rev. Finally, the preference values for spindle speeds are 6000 rpm better than 6500 rpm and 6500 rpm better than 7000 rpm. Refer to the rightmost column of Table 3, the ranking of the influence on tool life is A, B, C, and D. In other words, the importance of the four control factors to tool life in descending order is side clearance angle (A), coating layer (B), feed rate (C) and spindle speed(D).

It is evident that the analysis of Table 3 leads to the determination of an optimum parameter setting $A_1B_2C_3D_1$ causing minimum wear, that is, $A_1B_2C_3D_1$ is the best combination of parameters for achieving the desired cutting performance. However, from Table 3, the S/N ratio of A_2 is found to be 17.749 db, which is almost equals A_1 (17.914).

Table 3. Signal to noise response of flank wear measurement for micro-end-milling

Control Factor	Level 1	Level 2	Level 3	Max-Min	Ranking
A	17.914	17.749	13.731	4.813	1
B	15.716	18.518	15.164	3.354	2
C	14.983	16.498	17.912	2.929	3
D	17.685	16.120	15.589	2.096	4

The above analyses also suggest that there are interrelationships among the four aforementioned factors. Therefore, before any attempt is made to use this simple model as a predictor for measuring performance, the possible interactions between factors, AxB, AxC and BxC, must be considered. Thus, factorial design incorporates a simple means of testing for the presence of interaction effects was performed.

The S/N ratio response table and response graphs between factors A and B are shown in Table 4 and Fig. 5, respectively. Similarly, the response table and response graphs between factors A and C are shown in Table 5 and Fig. 6, respectively. The response table and response graphs between factors B and C are shown in Table 6 and Fig. 7, respectively.

Table 4. Interactions of S/N ratio between factors A and B

	A and B (S/N ratio)		
	B_1	B_2	B_3
A_1	16.90	19.66	17.19
A_2	16.15	22.47	14.62
A_3	14.08	13.42	13.68

Table 5. Interactions of S/N ratio between factors A and C

	A and C (S/N ratio)		
	C_1	C_2	C_3
A_1	16.90	19.66	17.19
A_2	14.62	16.15	22.47
A_3	13.42	13.68	14.04

Table 6. Interactions of S/N ratio between factors B and C

	B and C (S/N ratio)		
	C_1	C_2	C_3
B_1	16.90	16.15	14.08
B_2	13.42	19.66	22.47
B_3	14.62	13.68	17.19

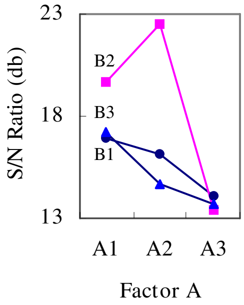


Fig. 5. Interactions of S/N ratio between control factors A and B

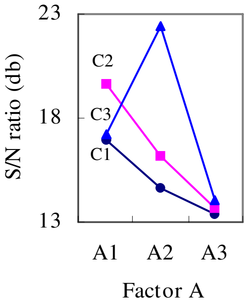


Fig. 6. Interactions of S/N ratio between control factors A and C

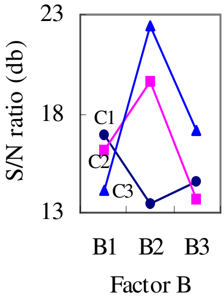


Fig. 7. Interactions of S/N ratio between control factors B and C

As can be seen from Figs. 5-7, the three factors are crossing, thus we can confirm that there are interrelations among factors A, B, and C. As a result, the following revision method is developed:

(1) As indicated in Table 4, the combination of A and B maximizing the S/N ratio is A_2B_2, its value is 22.47 db

(2) As can be seen from Table 5, the combination of A and C maximizing the S/N ratio is A_2C_3, its value is 22.47 db

(3) As shown in Table 6, the combination of B and C maximizing the S/N ratio is B_2C_3, its value is 22.47 db

Furthermore, on the basis of the above analyses, we can conclude that the best combination of factors A, B and C is $A_2B_2C_3$. The conclusion is drawn on the basis of the following truths:

- For factor A, since both (1) and (2) suggest A_2, we select A_2.
- For factor B, since both (1) and (3) suggest B_2, we select B_2.
- For factor C, since both (2) and (3) suggest C_3, we select C_3.

The last step of the Taguchi method is to verify the improvement of the control factors performance characteristic at the optimal levels of designed process parameters. The confirmation experiment is performed by conducting a new set of factor settings $A_2B_2C_3$ to predict the tool wear. The parameter condition for confirmation experiment (smaller-the-better analysis of tool wear) is shown in Table 7.

Table 7. Parameter condition for the confirmation experiment (smaller-the-better analysis of tool wear)

Recommend	Side Clearance Angle	Coating Layer	Feed Rate	Spindle Speed
level	A_2	B_2	C_3	D_1
Value	12^o	TiCN	0.0125	6000

3.2 Analysis of Variance

To see the effective parameters and their confidence levels on the milling of 6061 Al-alloy, statistical analysis of variance (ANOVA) were performed. The analysis results are shown in Table 8. With the performance characteristics and statistical analysis of variance (ANOVA), the optimal combination of machining parameters becomes predictable. As shown in Table 8, the side clearance angle is the most important control factor for tool life, followed by coating layer, feed rate, and spindle speed.

During the milling test, we fixed the milling width at 0.5 mm and cutting depth at 0.2 mm to examine the correlation between the side clearance angle and coating material. Moreover, among the four control factors explored, the side clearance angle plays the most important role. The four control factors are listed in descending order of importance as follows: side clearance angle, coating layer, feed rate and spindle speed. Their significances are 46.07%, 26.58%, 17.61%, and 9.74%, respectively. It is clear the contribution rate of spindle speed is relatively low.

Table 8. ANOVA Statistical Test Results

Sources of Variation	Degree of Freedom	Sum of Squares	Variance	Significance (%)
A	2	33.682	16.841	46.07
B	2	19.429	9.715	26.58
C	2	12.872	6.436	17.61
D	2	7.122	3.561	9.74
Total	8			100

4 Confirmation Experiment

The confirmation experiment is the final test in the design of the experiment process. The purpose is to validate the conclusions drawn during the analysis phase. The confirmation experiment is performed by conducting a new set of factor settings, for example A_2, B_2 and C_3, to predict the tool wear. The estimated S/N ratio for flank wear can be calculated with the help of the following prediction equation:

$$\eta = \overline{T} + (\overline{A}_2 - \overline{T}) + (\overline{B}_2 - \overline{T}) + (\overline{C}_3 - \overline{T}) + (\overline{D}_1 - \overline{T}) \tag{3}$$

where η denotes the predicted average value of the S/N ratio for flank wear; $\overline{T}$ denotes the mean of the 9 S/N ratios; $\overline{A}_2, \overline{B}_2, \overline{C}_3$ and $\overline{D}_1$ denote the mean responses of the four factors at the designated levels. By combining like terms, Eq. (2) reduces to

$$\eta = \overline{A}_2 + \overline{B}_2 + \overline{C}_3 + \overline{D}_1 \text{ - } 3\overline{T} \tag{4}$$

A new arbitrary combination of factor levels A_2, B_2 and C_3 is used to predict tool wear using the prediction equation and the S/N ratios is found to be:

$$\eta = \overline{A}_2 + \overline{B}_2 + \overline{C}_3 + \overline{D}_1 \text{ - } 3\overline{T} = (17.747) + (18.517) + (17.912) + (17.685) \text{ - } 3(16.460) = 22.48 \text{ (db)}$$

Here an experiment is conducted using the factor combination of A_2, B_2, C_3 and the result is compared with the value obtained from the predictive equation, as shown in Table 9. The resulting model seems to be able to predict tool wear to a reasonable accuracy.

Table 9. Results of the confirmation experiments for tool wear

Optimal control parameters	Prediction	Experimental
Recommend level	$A_2B_2C_3D_1$	$A_2B_2C_3D_1$
S/N ratio for tool wear (db)	22.48	22.39

5 Conclusions

The experiments were to mill 6061 Al-alloy using micro-end-mills. The object of this study is to examine the influence of side clearance angle, coating layer, feed rate, and spindle speed on flank wear. To quickly gain the interrelationships among these four control factors, we applied the Taguchi method to plan the experiments. The results could be useful in selecting a suitable combination of parameters for dry milling 6061 Al-alloy by using micro-end-mills and are summarized as follows:

1. The effect of edge detector and smoothing on the performance of the extract vertex algorithm has been explored in detail.
2. Mills wear can be on-line measurements of visual inspection technique based on the proposed method.
3. The best combination of the four control factors is $A_2B_2C_3D_1$. In other words, the optimum parameters for dry milling 6061 Al-alloy are 12 degrees side clearance angle, TiCN coating, 0.0125 mm/rev feed rate, and 6000 rpm spindle speed.

4. The most important cutting parameter is the side clearance angle, followed by coating layer feed rate, and spindle speed.
5. TiCN-coated mills generate less flank wear than the other two types of different-coatings.

References

1. Malamas, E.N., Petrakis, E.G.M., Zervakis, M., Petit, L., Legat, J.D.: A survey on industrial vision systems, applications and tools. Image and Vision Computing 21, 171–188 (2003)
2. Yoon, H.S., Chung, S.C.: Vision Inspection of Micro-Drilling Processes on the Machine Tool. Transactions of NAMRI/SME 32, 391–398 (2004)
3. Pfeifer, T., Wiegers, L.: Reliable Tool Wear Monitoring by Optimized Image and Illumination Control in Machine Vision. Measurement 28(3), 209–218 (2000)
4. Tsao, C.C.: Prediction of Flank Wear Different Coated Drills for JIS SUS304 Stainless Steel using Neural Network. J. Materials Processing Technology 123, 354–360 (2002)
5. Zhang, M.Z., Liu, Y.B., Zhou, H.: Wear Mechanism Maps of Uncoated HSS Tools Drilling Die-cast Aluminum Alloy. Tribology International 34, 727–731 (2001)
6. Nouari, M., Ginting, A.: Wear Characteristics and Performance of Multi-layer CVD-Coated Alloyed Carbide Tool in Dry End Milling of Titanium Alloy, Surface & Coatings Technology 200, 5663–5676 (2006)
7. Choi, Y.J., Chung, S.C.: Monitoring of Micro-Drill Wear by Using the Machine Vision System. Transactions of NAMRI/SME 32, 143–150 (2006)
8. Sreeram, S., Kumar, A.S., Rahman, M., Zaman, M.T.: Optimization of Cutting Parameters in Micro-end-milling Operations under Dry Cutting Conditions Using Genetic Algorithms. Int. J. of Advance Manufacture Technology 30, 1030–1039 (2006)
9. Jantunen, E.: Indirect Multisignal Monitoring and Diagnosis of Drill Wear. VTT Publications 590, Finland (2005)
10. Sreejith, P.S., Ngoi, B.K.A., Mater, J.: Process. Technol. 101 (2000)
11. Schulz, H., Dorr, J., Rass, I.J., Schulze, M., Leyendecker, T., Erkens, G.: Surface & Coating Technology 480, 146–147 (2001)
12. Yeo, S.H., Goh, K.M.: The Effect of Ultrasound in Micro Electro Discharge Machining on Surface Roughness. Proceedings of the Institution of Mechanical Engineers Part B-Journal of Engineering Manufacture 215, 271–276 (2001)
13. Kim, H.Y., Ahn, J.H., Kim, S.H., Takata, S.: Real-time drill wear estimation based on spindle motor power. J. Materials Processing Technology 124, 267–273 (2002)
14. Murali, M., Sundaram, Ganesh, B., Pavalarajan, Kamlakar, P.R.: Study on Process Parameters of Ultrasonic Assisted Micro EDM Based on Taguchi Method. Journal of Materials Engineering and Performance 17(2), 210–215 (2008)
15. Jantunen, E.: Indirect multisignal monitoring and diagnosis of drill wear. Technical Research Centre of Finland. VTT Publications, Espoo (2005)
16. Murali, M., Yeo, S.H.: Rapid Biocompatible Micro Device Fabrication by Micro Electro-Discharge Machining. Biomedical Microdevices 6, 41–45 (2004)
17. Ross, J.: Taguchi Techniques for Quality Engineering. McGraw-Hill, Inc., New York (1996)

Workflow Planning in Holonic Manufacturing Systems with Extended Contract Net Protocol

Fu-Shiung Hsieh and Chih Yi Chiang

Department of Computer Science and Information Engineering,
Chaoyang University of Technology,
41349 Taichung County, Taiwan, R.O.C.
fshsieh@cyut.edu.tw

Abstract. Holonic manufacturing systems (HMS) provide a reconfigurable and flexible architecture based on the notion of holon to dynamically meet customers' requirements. Given an order with specific product demands and due date, a challenge is to design a problem solving platform to guide the holons in HMS such that the decisions made by the individual holons as a whole fulfill the order. The objective of this paper is to propose a problem solving platform that finds a set of minimal cost holons and the associated workflows to fulfill an order. To achieve these objectives, we combine multi-agent systems technology with Petri nets to design and implement HMS to fulfill the requirements of orders. We first propose architecture and a two-layer contract net protocol for planning order holons, product holons and resource holons in HMS. To determine whether it is feasible for a set product holons and resource holons to meet the order requirements, we propose Petri net models to capture the workflows and activities in product holons and resource holons, respectively. To realize the two-layer contract net protocol, we proposed a scheme for publication and discovery of holon services based on FIPA compliant multi-agent system platform. We develop a HMS system to solve the order planning problem.

Keywords: Holonic manufacturing system, contract net, workflow.

1 Introduction

Holonic manufacturing systems (HMS) [1] is based on the notion of holon [7] to provide a reconfigurable and flexible manufacturing architecture to dynamically accommodate changes and meet customers' requirements. In HMS, a holon is an autonomous, co-operative and intelligent entity able to collaborate with other holon to process the tasks. An HMS consists of three types of holons: resource holons, product holons and order holons [12]. In existing literature, there are many studies on HMS [3], [4], [5], [9],[10], [11], [12], including flexible manufacturing planning [14], formal specification of the dynamic behaviour of holonic systems [6], [4] as well as agile and adaptive manufacturing control architecture [8]. A challenge is to design a platform to guide the holons such that the decisions made by the individual holons as a whole achieves the objectives such as meeting customers' orders and due dates. Although theoretical development on optimization of HMS has been studied in [5], implementation issues

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 701–710, 2009.

have not been addressed. The objective of this paper is to propose a problem solving platform that finds a set of minimal cost holons and the associated workflows to fulfill an order.

To achieve the objectives, we combine multi-agent systems technology with Petri nets in this paper to design and implement HMS. We first propose two-layer contract net protocol for planning order holons, product holons and resource holons in HMS. The two-layer contract net protocol consists of an upper layer protocol and a lower layer protocol. The upper layer is applied between an order holon and the potential product holons to minimize the costs and form a holarchy, where a holarchy is a set of holons to achieve an objective. The lower layer protocols applied between the holarchy formed in the upper layer protocol and the potential resource holons. To analyze the feasibility of the decisions made by individual holons, Petri net models [12], [13] are introduced. We propose Petri net models to capture the workflows and activities in product holons and resource holons, respectively. We develop the software called HMS Designer to facilitate the design of HMS. In HMS Designer, the design methodology is broken down into three steps. First, we construct workflow models for each product holons and resource activity models for each resource holon in Petri nets using PNML [2]. Second, we define each order holon, product holon and resource holon in the system using our Holon Editor. Third, we apply a two-layer contract net protocol between the order holon, product holons and resource holons to find a solution to fulfill the orders. We demonstrate the function of HMS Designer by an example.

The remainder of this paper is organized as follows. In Section 2, we formulate the workflow planning problem in HMS. In Section 3, we propose a two-layer contract net protocol for order holons, product holons and resource holons in HMS. In Section 4, we introduce a workflow model for each product holon. We also propose a resource activity model for each resource holon. In Section 5, we detail the implementation of the proposed concept. We conclude this paper in Section 6.

2 Workflow Planning in Holonic Manufacturing Systems

Fig. 1 shows our proposed architecture for workflow planning in HMS. An HMS consists of three types of holons: resource holons, product holons and order holons [1]. A resource holon consists of production resources with relevant components to control the resources. A product holon contains the production process information to manufacture products. An order holon represents an order. Given an order with specific product demands and due date, the problem is to find a set of holons in HMS to fulfill the orders. The HMS Designer aims to provide a multi-agent problem solving platform to model, construct and coordinate holons in HMS to fulfill an order. In Fig. 1, the workflow model repository and resource activity model repository captures the workflows of product holons and the activities of resource holons, respectively. An order manager, product manager and a resource manager are responsible for creation of an order holon, product holons and resource holons, respectively. To process an order, a set of resource holons and product holons form a composite holon called a holarchy. We formulate the problem to determine the best product holons and resource holons to fulfill an order as follows. Let $c(H, R)$ denote the holarchy formed by a set H of product holons and a set R of resource holons. Let $w: \{c(H, R)\} \rightarrow R^{+}$ be the cost function that specifies the cost of $c(H,R)$, where R^{+} denotes the set of positive real values. We formulate the following problem to minimize the cost while fulfill the order requirements.

Problem: $\min\limits_{R,H} w$ subject to the constraint that $c(H,R)$ is feasible and satisfies the time constraint, $R \subseteq \boldsymbol{R}$, $H \subseteq \boldsymbol{H}$, where $\boldsymbol{R}$ denotes the set of all potential resource holons and $\boldsymbol{H}$ denotes the set of all potential product holons.

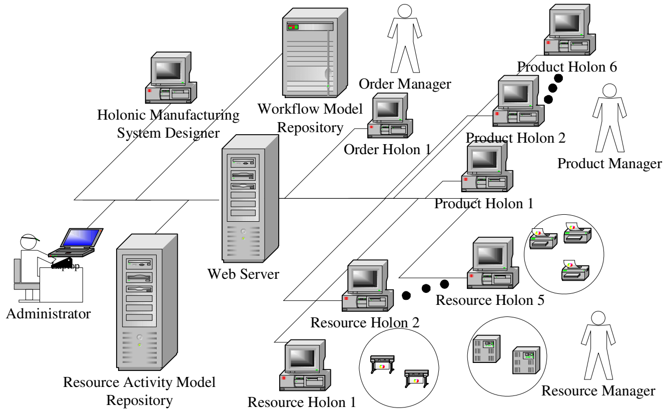


Fig. 1. Architecture of Holonic Manufacturing System

One way to form a holarchy is based on the contract net protocol (CNP) [15]. In CNP, there are two roles an agent can play: manager or bidder. Four stages are involved to establish a contract between a manager and one or more bidders: (1) Request for tender, (2) Submission of proposals, (3) Awarding of contract and (4) Establishment of contract: Formation of holonic processes in HMS based on CNP has been studied in [5], where CNP is applied to form a holarchy base on establishment of contracts between a set of holons. We propose a two-layer CNP which consists of an upper layer protocol and a lower layer protocol to select the holons required to fulfill an order. The upper layer protocol is applied between an order holon and the potential product holons to find the minimal cost product holons that form a holarchy. The lower layer protocol is applied between the holarchy formed by the upper layer protocol and the potential resource holons to analyze the feasibility and determine the resource holons.

3 Two-Layer Contract Net Protocol

Resource holons, product holons, and order holons are assigned different roles at different phases in the two-layer CNP. To describe the negotiation processes between an order holon, the product holons and the resource holons in HMS, the following messages are defined.

Definition 2.1. X_{rft} denotes a "Request for tenders" message sent by either an order holon or a product holon. X_{sop} denotes the "Submission of proposals" message sent by either a product holon or resource holon. X_{aoc} denotes the "Awarding of contract" message sent by an order holon or a product holon. X_{eoc} denotes the "Establishment of contract" message sent by an awarded product holon or an awarded resource holon.

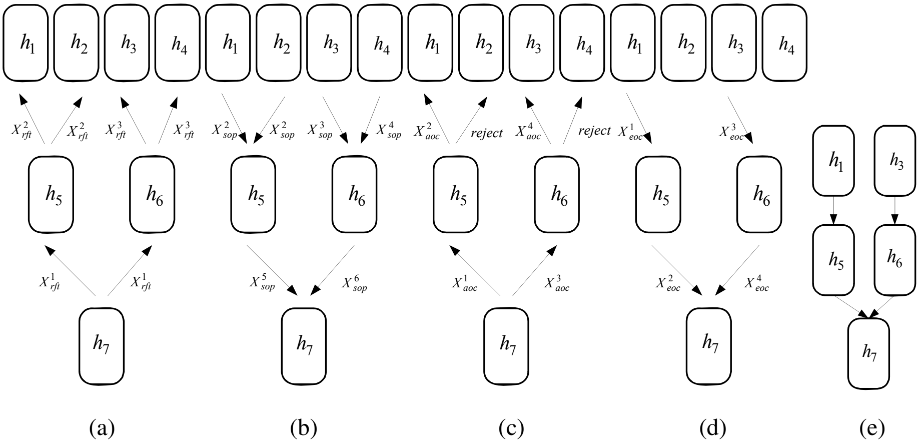


Fig. 2. (a) Request for tender (b) Submission of proposal (c) Awarding of contract (d) Establishment contract (e) A holarchy H

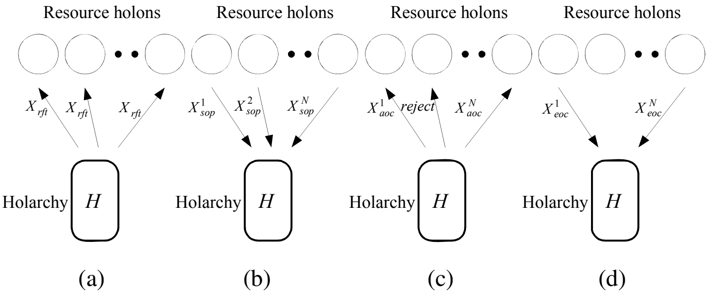

Resource holons
X_{rft} X_{rft} X_{rft}
X_{sop}^1 X_{sop}^2 X_{sop}^N
X_{aoc}^1 reject X_{aoc}^N
X_{eoc}^1 X_{eoc}^N
Holarchy H
(a) (b) (c) (d)

Fig. 3.

Figure 2(a)~(d) illustrate a scenario in which a production process is to be formed in HMS by applying the upper layer protocol between the potential product holons $h_1, h_2, ..., h_7$ to find the minimal cost product holons to fulfill an order holon o_1. Holonic processes are production processes dynamically created based on the collaboration of product holons. Each product holon has an internal process flow. Execution of the internal process of a product holon may rely on the outputs from the internal processes of one or more upstream product holons. For example, product holon h_7 depends on either h_5 or h_6 to provide the required parts. Furthermore, h_5 also depends on either h_1 or h_2 to provide the required parts whereas h_6 depends on either h_3 or h_4 to provide the required parts. To minimize the costs in the upper layer protocol, each product holon selects the minimal cost upstream product holons to establish the constracts. Figure 2(e) shows the holarchy H formed resulting from the upper layer protocol, where the cost of h_1 is less than that of h_2 and the cost of h_3 is less than that of h_4. The cost of a product holon is described by summing up the cost of the transitions in the associated Petri net model to be detailed in Section 4. Figure 3(a)~(d) show how the lower layer protocol is applied between a holarchy H and the potential resource holons $r_1, r_2, ..., r_N$. The contracts established among the set of product holons and resource holons form a holarchy to fulfill the orders.

4 Petri Net Representation of Holons

An important issue in implementation of HMS is to model the product holons and resource holons and represent the proposals submitted by them. Petri nets are powerful modeling tools to capture the internal workflows and processes of holons. Petri Net Markup Language (PNML) [2] is a standard for the representation of Petri nets. Therefore, we adopt Petri nets to model the internal processes of holons.

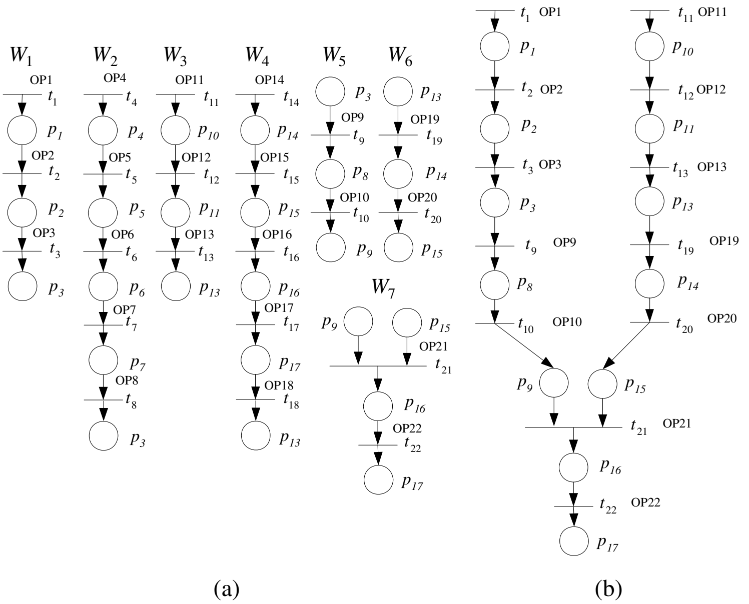

W_1
OP1 t_1 p_1 OP2 t_2 p_2 OP3 t_3 p_3
W_2
OP4 t_4 p_4 OP5 t_5 p_5 OP6 t_6 p_6 OP7 t_7 p_7 OP8 t_8 p_3
W_3
OP11 t_{11} p_{10} OP12 t_{12} p_{11} OP13 t_{13} p_{13}
W_4
OP14 t_{14} p_{14} OP15 t_{15} p_{15} OP16 t_{16} p_{16} OP17 t_{17} p_{17} OP18 t_{18} p_{13}
W_5
p_3 OP9 t_9 p_8 OP10 t_{10} p_9
W_6
p_{13} OP19 t_{19} p_{14} OP20 t_{20} p_{15}
W_7
p_9 p_{15} OP21 t_{21} p_{16} OP22 t_{22} p_{17}
t_1 OP1 p_1 t_2 OP2 p_2 t_3 OP3 p_3 t_9 OP9 p_8 t_{10} OP10
t_{11} OP11 p_{10} t_{12} OP12 p_{11} t_{13} OP13 p_{13} t_{19} OP19 p_{14} t_{20} OP20
p_9 p_{15} t_{21} OP21 p_{16} t_{22} OP22 p_{17}
(a)
(b)

Fig. 4. (a) Workflow Petri nets for product holons (b) Collaborative workflow Petri nets

We propose Petri net models for product holons and resource holons as follows. To model the workflow of a product holon in Petri net, we use a place to represent a state in the workflow while a transition to represent an event or operation that brings the workflow from one state to another. The workflow of a product holon h_n is modeled by a subclass of Petri nets called an acyclic marked graph $W_n = (P_n, T_n, I_n, O_n, m_{n0})$, where the set P_n of places denotes the production states whereas the set T_n of transitions denotes the operations. Fig. 4(a) shows the workflow Petri nets for the product holons in Fig. 2. We use $c_n : T_n \to R^+$ to denote the cost to fire a transition in T_n, where R^+ denotes the set of positive real values. The cost of holon w_n is $\sum_{t \in T_n} c_n(t)$.

Definition 4.1. The workflow of w_n is an acyclic marked graph W_n = $(P_n, T_n, I_n, O_n, m_{n0})$. As each transition represents a distinct operation in a task, $T_j \cap T_k = \Phi$ for $j \neq k$.

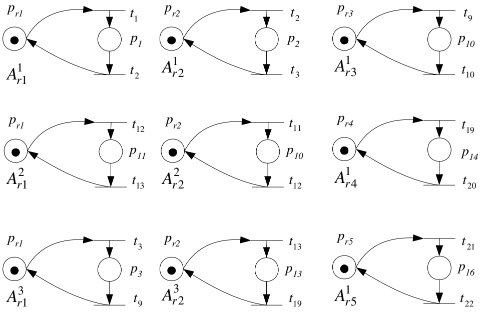


Fig. 5. Resource activities for the workflow in Fig. 4

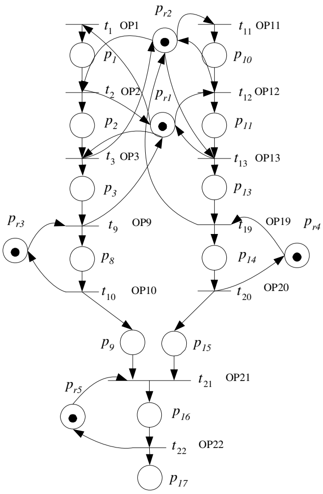


Fig. 6. A complete Petri net model

To carry out the operations for producing a final product relies on a set of product holons, we define the operator "$\|$" to merge two Petri net models with common transitions, places and/or arcs. Let N denote the set of product holons required for producing a final product. We define a collaborative workflow by $W = \|_{n \in N} W_n$. Fig. 4(b) shows a collaborative workflow Petri net $W = W_1 \| W_3 \| W_5 \| W_6 \| W_7$. Each operation in a workflow consumes a number of different types of resources. In addition to the workflow specified by W_n, the activities for each type of resources must also be specified. A resource may take part in some of the operations in different workflows and may visit different states in performing various operations in the workflows. A model is required to capture the activities of each type of resources. An activity is a sequence of operations to be performed and states to be visited by a certain type of resources. We use a place in Petri net to represent a state in the resource activity. Each resource has an idle state. Each resource activity starts and ends with an idle state. For example, consider the resource activity $A_{r1}^{1} = p_{r1}t_1p_1t_2$ for a resource in Fig. 5. Place p_{r1} denotes the idle state of type- r_1 resource. Firing transition t_1 changes the resource from idle state to a busy state. Resource activity $A_{r2}^{1} = p_{r2}t_2p_2t_3$ in Fig. 5 represents the activity of type- r_1 resource, where place p_{r2} denotes the idle state. Firing transition t_2 means starting OP2, which brings the resource from idle state to busy state. A resource activity is described by a circuit in Petri net. A circuit indicates that the resource activity includes resource allocation and de-allocation. Let $\boldsymbol{R}_n$ denote the set of resource types required to perform the operations in W_n. For the workflow W_1 in Fig. 4(a), $\boldsymbol{R}_1$ ={ $r_1, r_2, r_3, r_4, ..., r_{12}, r_{13}, r_{14}$ }. Fig. 5 shows the resource activities corresponding to the workflow in Fig. 6(a). The model for the $k-th$ activity of a type- r resources, where $r \in \boldsymbol{R}_n$, is described by a Petri net A_r^k as follows.

Definition 4.2. Petri net A_r^k = (P_r^k , T_r^k , I_r^k , O_r^k , m_r^k) denotes the $k-th$ activity for a type- r resources, where $r \in \boldsymbol{R}_n$. Remark that $T_r^k \cap T_r^{k'} = \Phi$ for $k \neq k'$.

Let K_r be the number of activities of a type- r resources. Let $\Omega_n^r \subseteq \{1,2,3,...,K_r\}$ denote the set of type- r activity IDs in W_n. The initial marking m_r^k is determined based on the set of resource tokens allocated to the $k-th$ activity. More specifically, $m_r^k(p_r)$ is the number of resources allocated to place p_r , where p_r is the idle place of type- r resources. A complete Petri net model C is modeled by combining the resource activity models with the workflow model to capture the interactions between the product holon and resource holons. To combine resource activity models with the workflow model, we define the operator "$\|$" as follows to merge two Petri net models with common transitions, places and/or arcs. C = $\|_{\gamma \in \boldsymbol{R}} A_\gamma \| W$, where A_γ = $\|_{k \in \Omega_n^\gamma} A_r^k$. Fig. 6 shows a complete Petri net model C = $A_{r1} \| A_{r2} \| A_{r3} \| A_{r4} \| A_{r5} \| W_1 \| W_3 \| W_5 \| W_6 \| W_7$.

Definition 4.3. A marking m is feasible if there exists a control policy u under which $C(m_0^*, u)$ is live.

Theorem 4.1. There exists a control policy u under which $C(m_0, u)$ is live if and only if the initial marking $m_0 \geq m_0^*$, where $m_0^* \in \mathrm{M}_0^*$ and M_0^* denotes the set of minimal feasible markings of C .

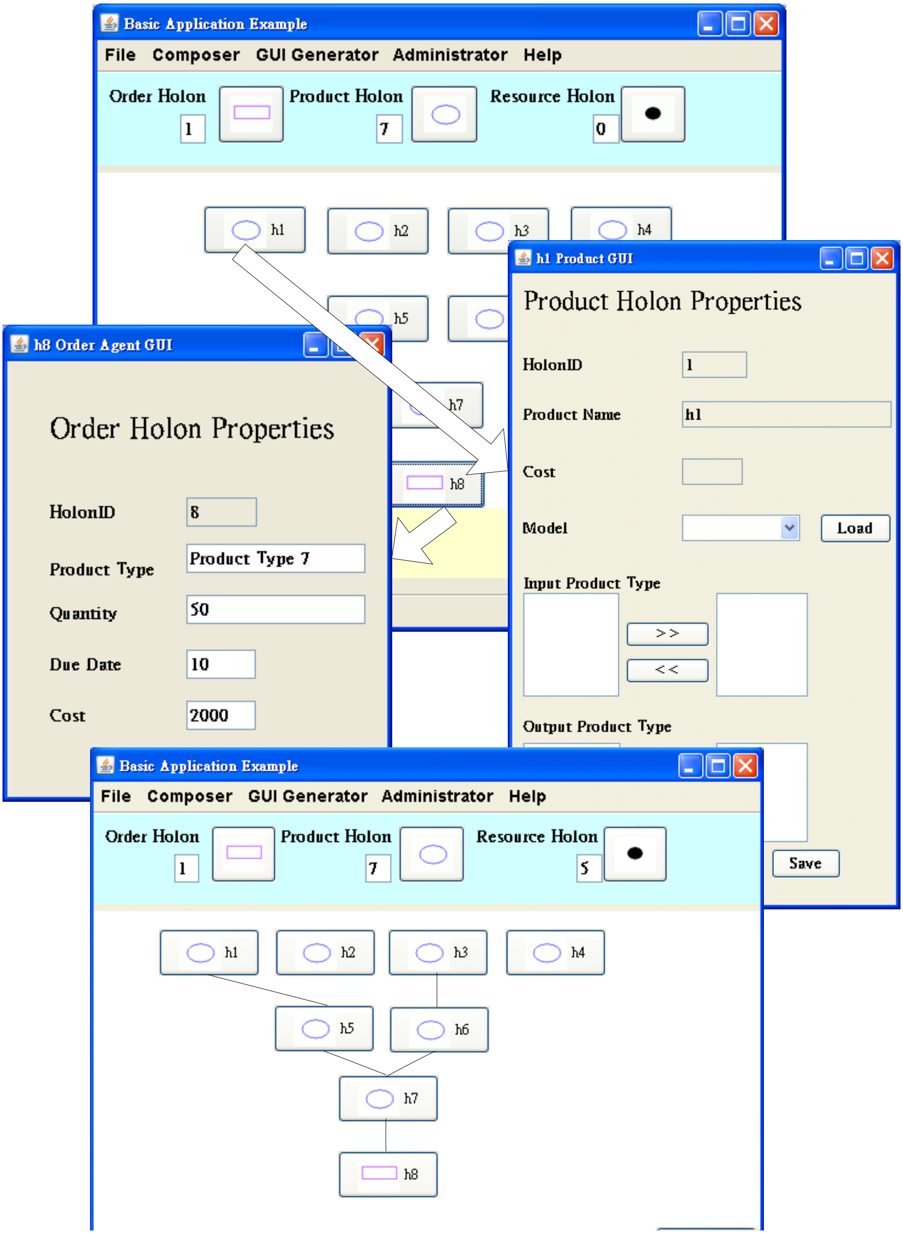
The computational complexity to determine a minimal feasible marking m_0^* of C is NP-Complete. The following theorem provides a upper bound of m_0^* .

Theorem 4.2. Given C , $m_0^*(p_r) \leq K_r \ \forall r \in \boldsymbol{R}_n$.

By combining Theorem 4.1 and Theorem 4.2, we determine whether the initial marking of C is feasible by checking the following conditions

$m_r^k(p_r) \geq K_r \ \forall r \in \boldsymbol{R}_n$.

If the above conditions hold, C is feasible.



Basic Application Example

Order Holon 1
Product Holon 7
Resource Holon 0

h1 Product GUI

Product Holon Properties

HolonID 1
Product Name h1
Cost
Model
Input Product Type
Output Product Type

h8 Order Agent GUI

Order Holon Properties

HolonID 8
Product Type Product Type 7
Quantity 50
Due Date 10
Cost 2000

Basic Application Example

Order Holon 1
Product Holon 7
Resource Holon 5

Fig. 7. Holonic Manufacturing System (HMS) Designer

5 Implementation

The two-layer contract net protocol relies on an infrastructure for individual holons to publish and discover their services. The Java Agent Development Environment (JADE) multi-agent platform provides a built-in directory service to simplify publishing and discovery of services. Each type of holons can be implemented as an agent in JADE. Messages exchanged by JADE agents is specified by the ACL language defined by FIPA. Each time a holon is added to the system, its services are published through the DF (Directory Facilitator) agent in JADE. To apply the contract net protocol, a manager must be able to search the DF agent for the services provided by the other bidders.

Fig. 7 shows the screen shot of our system, which consists of a HMS Editor and Monitor to define holons and monitor the interactions between holons. The following example details an application scenario of the HMS Designer.

Example: Consider the requirements of an order: required product: product type 7, quantity: 50, due date:10, and cost constraint: 2000. The upper part of Fig. 7 demonstrates the HMS Editor in which users (order holon managers, product holon managers and resource holon managers) may define the holons in a system. For this example, there are eight holons that have been defined, including seven product holons and one order holon. User may drag and drop the order holon, product holon and resource holon icons to define the holons. The left of Fig. 7 shows the graphical user interface for order holon managers to set order holon properties. The properties of an order holon include holon id, product type, quantity, due date and cost constraint. In Fig. 7, the product type, quantity, due date and cost is set to Product Type 7, 50, 10 and 2000, respectively. The right of Fig. 7 illustrates the graphical user interface for product holon managers to set product holon properties. The properties of a product holon include holon id, model, input/output product types and costs, where input product types is the product types required for producing the output product types. The contracts between product holons are shown in the lower part of Fig. 7.

6 Conclusion

Planning of workflows in HMS is a challenging issue. We propose a workflow planning methodology to fulfill order requirements in HMS by combining multi-agent systems technology with Petri nets. We first propose a two-layer contract net protocol (CNP) for planning order holons, product holons and resource holons in HMS. The upper layer protocol is applied between an order holon and the product holons whereas the lower layer protocol is applied between the holarchy formed by the upper layer and the resource holons. We propose Petri net models in PNML to capture the workflows and activities for product holons and resource holons, respectively. We proposed a scheme for publication and discovery of holon services based on JADE. This paper is differentiated from the works of W.M.P. van der Aalst [14], [15] in that we concentrate on workflow planning in HMS based on Petri net models and a two-layer CNP. The two-layer CNP proposed in this paper is differentiated from the work of Leitão *et al.* [9] which introduces the concept of autonomy factor and a pheromone-like spreading mechanism to deal with changes. This paper is different from the ones in [5] as it concentrates on design and implementation issues of HMS instead of theoretical development. We develop a software called HMS Designer to facilitate the design of HMS. In HMS Designer, users define each order holon, product holon and resource holon in the system using our Holon Editor and then apply a two-layer CNP between the order holon, product holons and resource holons to find a solution to fulfill the orders while avoiding deadlocks. We demonstrate the function of HMS Designer.

Acknowledgement

This paper is supported in part by National Science Council of Taiwan under Grant NSC97-2410-H-324-017-MY3.

References

1. Balasubramanian, S., Brennan, R.W., Norrie, D.H.: An architecture for metamorphic control of holonic manufacturing systems. Computers in Industry 46, 13–31 (2001)
2. Billington, J., Christensen, S., van Hee, K.M., Kindler, E., Kummer, O., Petrucci, L., Post, R., Stehno, C., Weber, M.: The petri net markup language: Concepts, technology, and tools. In: van der Aalst, W.M.P., Best, E. (eds.) ICATPN 2003. LNCS, vol. 2679, pp. 483–505. Springer, Heidelberg (2003)
3. Gou, L., Luh, P.B., Kyoya, Y.: Holonic manufacturing scheduling: architecture, cooperation mechanism, and implementation. Computers in Industry 37, 213–231 (1998)
4. Hsieh, F.S.: Analysis of contract net in multi-agent systems. Automatica 42(5), 733–740 (2006)
5. Hsieh, F.S.: Holarchy Formation and Optimization in Holonic Manufacturing Systems with Contract net. Automatica 44(4), 959–970 (2008)
6. Johnson, C.A.: Towards a formalized HMS model. In: Deen, S.M. (ed.) Agent-Based Manufacturing: Advances in the Holonic Approach, pp. 193–206. Springer, Heidelberg (2003)
7. Koestler, A.: The Ghost in the Machine. Hutchinson, London (1967)
8. Leitão, P., Restivo, F.: ADACOR: A holonic architecture for agile and adaptive manufacturing control. Computers in Industry 57(2), 121–130 (2006)
9. McFarlane, D.C., Bussmann, S.: Developments in holonic production planning and control. International Journal of Production Planning and Control 11(6), 522–536 (2000)
10. Neligwa, T., Fletcher, M.: An HMS operational model. In: Deen, S.M. (ed.) Agent-Based Manufacturing: Advances in the Holonic Approach, pp. 163–191. Springer, Berlin (2003)
11. Ramos, C.: A Holonic Approach for task scheduling in manufacturing systems. In: Proceedings of the 1996 IEEE International Conference on Robotics and Automation, pp. 2511–2516 (1996)
12. van der Aalst, W.M.P., Kumar, A.: A reference model for team-enabled workflow management systems. Data & Knowledge Engineering 38(3), 335–363 (2001)
13. van der Aalst, W.M.P.: The application of Petri nets to workflow management. J. Circuit. Syst. Comput. 8(1), 21–66 (1998)
14. Wullink, G., Giebels, M.M.T., Kals, H.J.J.: A system architecture for holonic manufacturing planning and control (EtoPlan). Robotics and Computer-Integrated Manufacturing 18(3-4), 313–318 (2002)
15. Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. IEEE Trans. On Computers 29, 1104–1113 (1980)

Machine Vision-Based Automatic Raw Fish Handling and Weighing System of Taiwan Tilapia

Yu-Teng Liang[1,2] and Yih-Chih Chiou[1]

[1] Institute of Engineering Science, Chung-Hua University,
No. 707, Sec.2, WuFu Rd., Hsinchu, 300, Taiwan
chiou@chu.edu.tw

[2] Department of Automation Engineering, Ta Hwa Institute of Technology,
No.1, Da-Hua Rd., Qiong-Lin, Hsinchu 30740, Taiwan
lyd337@thit.edu.tw

Abstract. This study proposes a vision-based automatic raw fish handling system to speed up fish cleaning and weighing. The proposed fish weighing system used a camera to capture projected images of fishes. Applying image processing techniques, physical properties of fishes, such as length, width, perimeter and area were obtained. Followed by regression analysis, weight-length, weight-height, weight-perimeter and weight-area relationships were derived. Analysis results of fifty tilapias show that coefficient of determination of the regression equation relating weight and area is 0.9303. The high value suggests that a tilapia's weight is highly correlated with its projected area. Therefore, use a tilapia's area to estimate its weight is justifiable.

Keywords: Machine Vision, Raw Fish Handling, Fish Weighing, Regression Analysis, Tilapia.

1 Introduction

For some years now, there have been both an increased demand for, and value attributed to higher quality of Taiwan tilapia. Tilapia of the 21st century is a valuable food fish. Typical demands include high economic worth and high protein content; it becomes a major source of animal proteins for human being. In view of that, the food and agriculture organization of the united nation is currently promoting the farming of tilapia intensively. Thus, we can foresee that farmed tilapia will play an important role in food fishes.

Machine vision, an important technique for inspection, measurement, sorting, remote sensing, surveillance, etc., has been extensively used in manufacturing and food industries. In addition, machine vision technique also has been used in agriculture and fishery industries for sizing, counting, weighing, grading, recognition, classification, and monitoring [1-3]. During biology study physical properties, such as length, width, thickness, area, weight, volume, perimeter, compactness and circularity are commonly used as a basis for sizing [4-7], weighing [8-10], grading [11, 13], or classification [14-16]. Martinez presents new low-cost systems for the automation of some fish farm operations. Particularly, computer vision is applied to non-contact fish

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 711–720, 2009.

weight estimation. White Trials of a computer vision machine (The Catch Meter) for identifying and measuring different species of fish. Shieh and Petrell adopted stereographic video technique to size salmons. Buckingham used machine vision technique to cut fish head off.

Tilapias are sold by their weights; as a result, it is necessary to develop a quick and accurate weighing method. Currently fish weighting is carried out manually using a weighing scale. Manual weighing is indeed an accurate means, but it is manual and time-consuming. To automate fish weighing, Mathiassen describe a proof-of-concept prototype of an automated system for weight and quality grading of pelagic fish using a multi-modal machine vision system combined with robotized sorting. Omar presents an optimal portion control technique for food processing which is being developed depends on fast on-line measurement of the weight distribution of each incoming food item. Line developed three models to estimate fish mass using truss lengths. Although much work has been done on examining the weight-length relationship of salmons, little attention has been devoted to explore the weight-area relationship. More importantly, tilapia is the most popular fish of cultivating kind in Taiwan. Therefore, the purpose of the study was to investigate the relationship between weight and area of tilapias using regression analysis and then use the derived relationship to establish a quick and accurate vision-based fish weighing system.

2 Materials and Methods

2.1 Fish and Fish Sampling

The experimental samples consisted of 150 tilapias having an overall length of between 25cm to 35cm and a weight of between 300g to 700g. All tilapias were uniform in shape and kindly provided by Guang-Ji fresh market. The samples were randomly divided into two groups. The first group consisted of 50 tilapias was used to explore weight-length, weight-height, weight-perimeter and weight-area relationships. The second group consisted of 110 tilapia was mainly used to verify the relationships established by the first group. Fig. 1 shows an ordinary tilapia sold in fresh market.

Fig. 1. An ordinary tilapia sold in a fresh market

2.2 Hardware and Software

Fig. 2 shows the schematic diagram of the proposed automatic fish handling system. Although the system consists of two major sub-systems: raw fish cleaning system (consists of a input slot ①, a processing platform ②, a output slot ③, a group of fish scaler ④, a group of conveyer belts ⑤, a rotating knife ⑥ Machine structure ⑦, a Cistern ⑧.) and machine vision-based weighing system. The fluorescent illuminator has a back lighting through which the camera can acquire image for measuring the physical properties of the fish. The physical properties used in the study include body-length, height, perimeter, area, weight and relationships among these; however, we will devote our discussion mainly to the research and development of the machine vision-based weighing system. Here, a computer vision algorithm was designed to extract the geometrical features of tilapia.

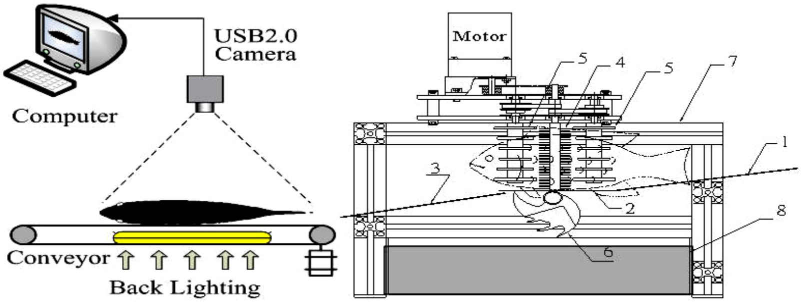


Fig. 2. Schematic diagram of the automatic raw fish handling system

2.3 Raw Fish Cleaning System

The major functions of the raw fish cleaning system include scaling, gutting, and rinsing. Scaling is a process to scrape the scales off the tilapias by using a fish scaler. Gutting is a process to open tilapia's belly using a sharp rotating knife and then pull out its innards (take out viscera). The rinsing process is to wash tilapias, knife, and platform using a large amount of water. As can be seen, the system was controlled by a PLC consisting of a CPU, an input module, and an output module. The input module is responsible for monitoring the signals from proximity sensors, optical sensors, or limit switches. The output module is responsible for turning on the conveyor, fish scaler, rotating knife and drain valves.

At the beginning, a raw fish is manually put into the sliding slot. Then the fish slides into the processing platform by its own weight. Immediately after optical sensor detects the presence of the fish, the system starts automatically. During its transportation by means of a conveyor, the fish is scaled off by a descale mechanism. After that, the fish is further transported by a set of transportation wheels. While the fish is transporting, a rotating knife cuts a slit from its anal opening to the gill openings and the viscera are pulled out. As the fish moves on, another optical sensor is triggered and a water spraying system is actuated subsequently to flush the knife and the platform. Finally, the fish slides down through the output slot to the vision-based weighing system. While the fish is sliding, it is thoroughly flushed by another water spraying system; Fig. 3 shows the processing flow of the automatic raw fish handling system.

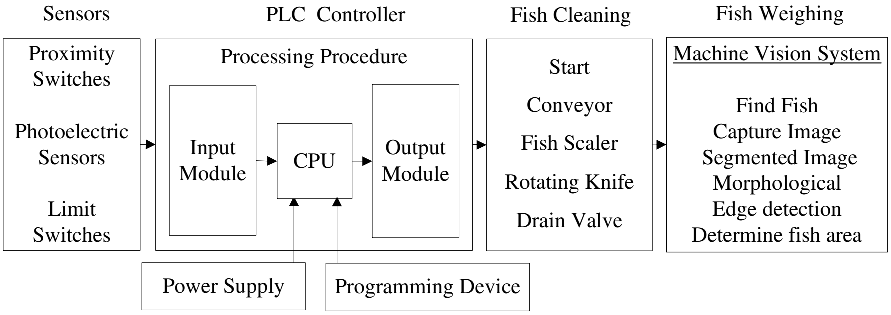


Fig. 3. Processing flow of the raw fish handling system

2.4 Vision-Based Weighing System

Referring to Fig.2, the vision-based weighing system was installed right after the raw fish cleaning system. The machine vision system consists of a CMOS camera, a lens, a fluorescent ring light, and a light controller. The 1280×1024 USB2.0 color CMOS camera from ARTRAY Japan was used to capture tilapias' grayscale image of size 512×480. The FCL-30D fluorescent ring light was used for illumination. The ring light has a diameter of 29 mm and provides a maximum lumen of 1400 Lm. The color temperature is 6200K. The light controller was used to adjust the intensity of the ring light. In the research, a USAF glass slide resolution target from Edmund Optics was used in calibration to derive the scale factor (μm/pixel, mm/pixel, cm/pixel, etc) of the vision system. The functioning of a general machine vision-based weighing system can subdivide into several steps, as presented in Fig 4.

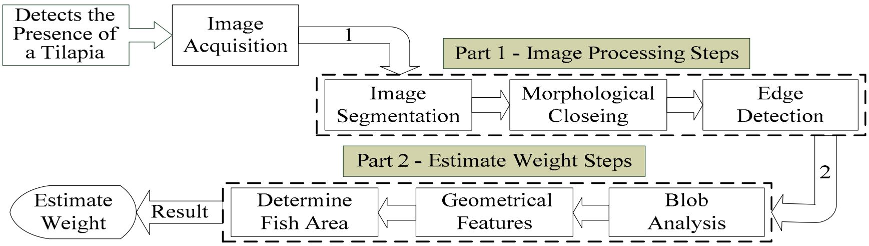


Fig. 4. Model of a typical machine vision-based weighing system: part 1 - image processing steps; part 2 - estimate weight steps

2.5 Experimental Method

The experiment is carried out in three phases: In the first phase, the tilapias in both groups were measured manually to obtain their weights, lengths, heights, perimeters and areas. Then data from the first group were analyzed using linear regression techniques to build the relationships between weight and physical properties. The objective of the first phase is to see which relationship is better correlated. In the second phase, we used the vision-based weighing system to obtain the weights of the tilapias in the first group. More specifically, we applied machine vision techniques to acquire the physical properties of each tilapia in the first group and used these data to estimate fish weights. Finally, the weights obtained by the two methods were compared to reveal the performance of the vision-based weighing system.

Acquire Physical Properties of Tilapias. The physical properties used in the study include length, height, perimeter, area and weight. Both width and height were measured using a vernier caliper. In the study, we used a weighing scale to obtain a tilapia's weight. A fish's area and perimeter was placed on a regular graph paper and its profile was drawn. Then the fish's area was obtained by accumulating the area enclosed by the profile of the fish. Here, we defined the length of a tilapia as the horizontal distance between the tip of its mouth and the farthest point on the caudal fin. The height of a tilapia is defined as the vertical distance between the highest point of the dorsal fin and lowest point of the ventral fin. Table 1 shows only the areas and the weights of the 50 tilapias in the first group.

Table 1. Manually measured weights (W) and areas (A) of the 50 tilapias in the first group

No.	W(g)	A(cm^2)	No.	W(g)	A(cm^2)	No.	W(g)	A(cm^2)	No.	W(g)	A(cm^2)
1	388	230.30	14	570	274.25	27	490	242.90	40	638	296.00
2	428	234.06	15	492	253.58	28	480	252.60	41	602	273.70
3	352	220.60	16	496	244.89	29	512	255.20	42	600	266.00
4	374	227.70	17	462	248.10	30	460	233.91	43	616	273.30
5	450	251.07	18	436	237.16	31	484	248.40	44	584	285.30
6	462	245.40	19	432	231.26	32	528	254.20	45	482	251.35
7	332	214.10	20	464	250.66	33	498	2420	46	454	240.08
8	338	208.20	21	418	235.76	34	442	239.9	47	484	242.99
9	444	238.35	22	442	234.70	35	610	292.10	48	484	246.30
10	388	231.8	23	408	224.60	36	746	309.80	49	440	230.10
11	408	216.77	24	428	232.06	37	678	296.20	50	388	224.29
12	388	231.08	25	558	270.70	38	574	265.95			
13	360	214.10	26	512	268.30	39	684	308.10			

Establish Regression Models. Once required data are carefully measured, the next step is to use data of the first group to establish the relationships between weight physical properties. In the study, we assumed that there are linear relationships between weight and length, height, perimeters and area. The relationships are governed by the following equation:

$$y = mx + b \tag{1}$$

where x represents the length, the height, the perimeter or the area of a tilapia; y denotes the weight of the tilapia; m and b are the slope and the y-intercept of the best-fit straight line, respectively.

(a) The Relationship between Weight and Length. The relationship between weight and length can be derived by using a scatter diagram. It can be seen from Fig. 5 that there exists a positive correlation between tilapia weight (*W*) and tilapia length (*L*). Through regression analysis, we can identify the exact relationship. As we assume that the relationship between weight and length is linear, the best-fit linear regression equation is shown in Eq. (2). It is clear that a tilapia's weight increases with its overall length. The symbol R^2 is Coefficient of Determination (abbreviated as CoD). Actually, the value of CoD provides us a measure of the strength of the linear relationship between weight and length of tilapias.

$$W = 59.056 * L - 1305.3 \tag{2}$$

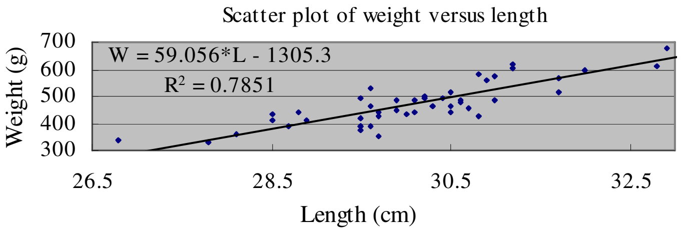


Fig. 5. The linear relationship between the weight (W) and the length (L) of tilapias

(b) The Relationship between Weight and Height. Similarly from the linear regression analysis results of the data collected from the first group, we found that there is a positive correlation between the weight (W) and height (H) of a tilapia. Fig. 6 shows the scatter plot of the 40 tilapia and the derived linear regression line. The linear regression equation relating weight and height is shown in Eq. (3). The R^2 value is 0.791. As you can see from the plot, a tilapia's weight also increases linearly with its height.

$$W = 86.995 * H - 538.26 \tag{3}$$

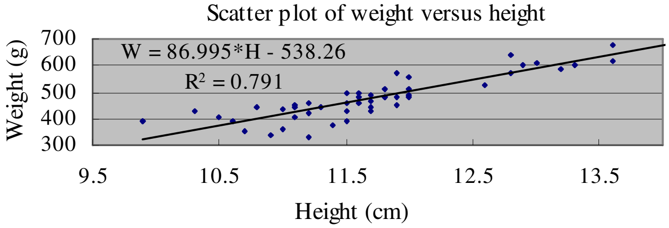


Fig. 6. The linear relationship between the weight (W) and the height (H) of tilapias

(c) The Relationship between Weight and Perimeter. The relationship between weight and perimeter can be derived by using a scatter diagram. It can be seen from Fig. 7 that there exists a positive correlation between tilapia weight (*W*) and tilapia perimeter (*P*). Through regression analysis, we can identify the exact relationship. As we assume that the relationship between weight and perimeter is linear, the best-fit linear regression equation is shown in Eq. (4). The R^2 value is 0.8216.

$$W = 21.663 * P - 1351.4 \tag{4}$$

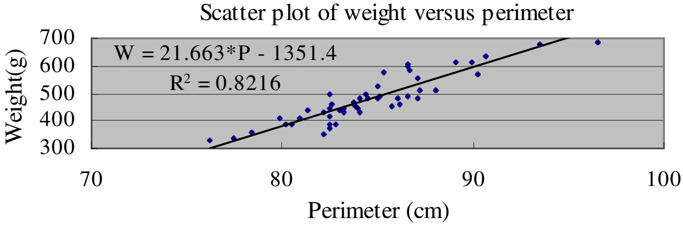


Fig. 7. The linear relationship between the weight (W) and the perimeter (P) of tilapias

(d) The Relationship between Weight and Area. Based on the linear regression analysis result, we found that there is a positive correlation between the weight (*W*) and the area (*A*) of a tilapia. Fig. 8 shows the plot of tilapia weight versus tilapia area. The linear regression equation relating weight and area is shown in Eq. (5).

$$W = 3.7074 * A + 438.66 \tag{5}$$

It is evident that the weight of a tilapia increases with its area. The R^2 value is 0.9722 indicating that 97.22% of the variation in the weight of a tilapia may be explained by its area. Since CoD is very close to one, it indicates that an excellent linear reliability exists between the weight and the area of a tilapia.

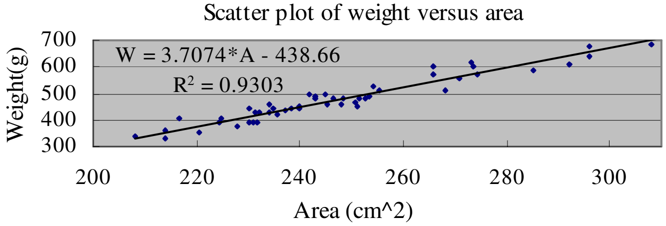


Fig. 8. The linear relationship between the weight (W) and the area (A) of tilapias

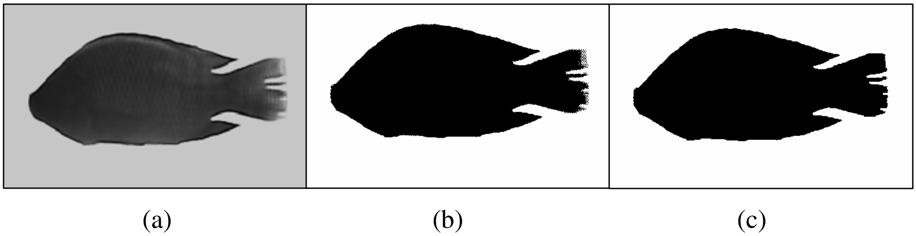
(e) Choose a Regression Model for Weighing Tilapias. From the results as shown in Figs. (5) - (8), we found that a tilapia's weight is linear correlated with its length, height, perimeter and area. However, a tilapia's area has a greater influence on its weight. In other words, a tilapia's weight is more closely related to its area than its perimeter, length or height. As a result, we decided to use the relation between weight and area to weigh tilapias. In summary, a tilapia's weight was estimated by plugging the area acquired by the machine vision-based weighing system into Eq. (5). Of course, a tilapia's area will be determined by image processing techniques that will be described in detail in the following section.

3 Results and Discussions

3.1 Weigh by Machine Vision

The object of the current step is to collect area of the individual tilapias in the second group. Once each tilapia's projected area has been obtained using the vision-based weighing system, we then made use of the regression equation as shown in Eq.(5) to evaluate its weight. For ease of understanding, the processing flow is divided into the following four steps:

(1) Segment the fish image as shown in Fig. 9a using Otsu's thresholding method to obtain the resulting binary image shown in Fig. 9b.
(2) Apply morphological closing operation to the binary image to fill small holes and smooth the profile of the fish. Fig. 9c shows the resulting image.
(3) Determine fish area using blob analysis technique.
(4) Plugging a tilapia's area into Eq. (5) to determine its estimated weight.



(a) (b) (c)

Fig. 9. Image processing steps for the weighing of raw fishes: (a) grabbed image of a tilapia using a backlighting illumination; (b) Segmented image using Otsu's thresholding method; (c) Resulting fish image after morphological closing operation

3.2 Comparison Results and Weighing Error Analysis

The system is capable of processing 15 to 20 tilapias per minute. To verify the accuracy of the vision-based weighing system, the weighing results of the second group derived in Sections Acquire Physical Properties of Tilapias and Weigh by machine vision respectively were compared. Table 2 shows a partial list of the comparison results, where A is the area determined by vision system, W_1 is the weight obtained manually using a weigh scale, W_2 is the weight evaluated by the regression equation shown in Eq. (5), and ε is the difference between the regressed fish weight and the actual fish weight. For simplicity, let W_1 be the actual fish weight; W_2 be the regressed fish weight; and ε be the weighing error. Thus $\varepsilon = W_1 - W_2$.

As can be seen from Table 2, the maximum weighing error, mean weighing error, and standard deviation of the weighing error are 9.12g, 1.77g, and 2.21g, respectively. Moreover, the percentage of the maximum error and mean error of the weighing is 1.97% and 0.83%, respectively. It is worth mentioning that the low weighting errors indicate that we can estimate a tilapia's weight by its area with a maximum error of 1.97% of its actual weight.

Table 2. Area (A), actual weight (W_1), regression weight (W_2) of tilapias in the data set

No.	A (cm^2)	W_1(g)	W_2(g)	ε (g)	ε %	No.	A (cm^2)	W_1(g)	W_2(g)	ε (g)	ε %
1	238.18	442	444.35	-2.35	0.53	26	229.86	408	413.52	-5.52	1.35
2	228.93	408	410.06	-2.06	0.50	27	223.53	388	390.04	-2.04	0.53
3	232.81	428	424.46	3.54	0.83	28	213.97	360	354.62	5.38	1.49
4	268.24	558	555.82	2.18	0.39	29	223.66	388	390.52	-2.52	0.65
5	256.07	512	510.70	1.30	0.25	30	233.38	428	426.57	1.43	0.33
6	244.83	462	469.02	-7.02	1.52	31	214.35	352	356.02	-4.02	1.14
7	208.04	332	332.62	-0.62	0.19	32	271.03	570	566.15	3.85	0.68
8	209.42	338	337.75	0.25	0.07	33	250.95	492	491.72	0.28	0.06
9	237.50	444	441.84	2.16	0.49	34	302.08	678	681.26	-3.26	0.48
10	223.90	388	391.42	-3.42	0.88	35	272.21	574	570.52	3.48	0.61
11	251.62	490	494.21	-4.21	0.86	36	219.83	374	376.33	-2.33	0.62
12	248.92	480	484.19	-4.19	0.87	37	248.18	484	481.46	2.54	0.52
13	254.25	512	503.96	8.04	1.57	38	236.43	440	437.89	2.11	0.48
14	275.26	584	581.84	2.16	0.37	39	222.62	388	386.67	1.33	0.34
15	248.89	482	484.07	-2.07	0.43	40	245.94	464	473.12	-9.12	1.97
16	242.18	454	459.19	-5.19	1.14	41	233.04	418	425.33	-7.33	1.75
17	250.58	484	490.35	-6.35	1.31	42	240.95	460	454.65	5.35	1.16
18	254.09	498	503.35	-5.35	1.07	43	247.60	484	479.28	4.72	0.98
19	236.20	442	437.04	4.96	1.12	44	262.96	528	536.24	-8.24	1.56
20	284.78	610	617.12	-7.12	1.17	45	279.96	602	599.27	2.73	0.45
21	320.09	746	748.06	-2.06	0.28	46	281.14	600	603.65	-3.65	0.61
22	254.47	496	504.77	-8.77	1.77	47	285.70	616	620.53	-4.53	0.74
23	241.99	462	458.48	3.52	0.76	48	304.12	684	688.83	-4.83	0.71
24	236.70	436	438.89	-2.89	0.66	49	289.38	638	634.19	3.81	0.60
25	233.59	432	427.35	4.65	1.08	50	238.07	450	443.95	6.05	1.34

4 Conclusions

This paper has presented a new method and an apparatus for automatically cleaning and weighing of tilapias. On the basis of the regression analysis results, we discovered that a tilapia's weight is highly correlated with its projected area. Thus, it is reasonable to use a tilapia's projected area to estimate its weight. The research results might help fish provider, especially fresh marketer, reducing the highly labor intensity of fish cleaning and weighing. Currently a fish weight is derived using only its area without considering its thickness. In the future, we will incorporate thickness into analysis. Hopefully, the weighing accuracy can be increased.

Acknowledgments. We thank Guang-Ji fresh market for providing tilapias that make the study possible.

References

1. Martinez-de Dios, J.R., Serna, C., Ollero, A.: Computer Vision and Robotics Techniques In Fish Farms. Robotica 21(3), 233–243 (2003)
2. White, D.J., Svellingen, C., Strachan, N.J.C.: Automated measurement of species and length of fish by computer vision. Fisheries Research 80, 203–210 (2006)

3. Misimi, E., Mathiassen, J.R., Erikson, U.: Computer Vision-Based Sorting of Atlantic Salmon (Salmo salar) Fillets According to Their Color Level. Journal of Food Science 72(1), s30–s35 (2007)
4. Naiberg, A., Petrell, R.J., Savage, C.R., Neufeld, T.P.: Stereo Video Technique to Size Fish in Sea Cage and Tanks. Aquaculture Engineering, 393–402 (1993)
5. Petrell, R.J., Shi, X., Ward, R.K., Naiberg, A., Savage, C.R.: Determining Fish Size and Swimming Speed in Cages and Tanks Using Simple Video Techniques. Aquaculture Engineering 16, 63–83 (1997)
6. Shieh, A.C.R., Petrell, R.J.: Measurement of Fish Size in Atlantic Salmon (salmo salar 1.) Cages Using Stereographic Video Techniques. Aquaculture Engineering 17, 29–43 (1998)
7. Misimi, E., Mathiassen, J.R., Erikson, U., Skavhaug, A.: Computer vision based sorting of Atlantic salmon (Salmo salar) according to shape and size. In: Proceedings of VISAPP International Conference on Computer Vision Theory and Applications, Setubal, Portugal, pp. 265–270 (2006)
8. Omar, F., Silva, C.W.D.: High-speed model-based weight sensing of complex objects with application in industrial processes. Measurement 33, 23–33 (2003)
9. Lines, J.A., Tillett, R.D., Ross, L.G., Chan, D., Hockaday, S., McFarlane, N.J.B.: An Automatic Image-Based System for Estimating the Mass of Free-Swimming Fish. Computers and Electronics in Agriculture 31, 151–168 (2001)
10. Bailey, D.G., et al.: Three-dimensional Vision for Real time Produce Grading. In: Harding, K.G., Miller, J.W.V. (eds.) Proceedings of SPIE, Machine Vision and Three-Dimensional Imaging Systems for Inspection and Metrology II, vol. 4567, pp. 171–178 (2002)
11. Mathiassen, J.R., Jansson, S., Veliyulin, E., Njaa, T., Lonseth, M., Bondo, M., Østvik, S.O., Risdal, J., Skavhaug, A.: Automatic Weight And Quality Grading Of Whole Pelagic Fish. In: NFTC 2006 Nor-Fishing Technology Conference, Trondheim, Norway, pp. 1–8 (2006)
12. Croft, E.A., de Silva, C.W., Kurnianto, S.: Sensor Technology Integration in An Intelligent Machine for Herringroe Grading. IEEE/ASME Transactions on Mechatronics 1(3), 204–215 (1996)
13. Brosnan, T., Sun, D.W.: Inspection and grading of agricultural and food products by computer vision systems – a review. Computers and electronics in agriculture 36(2-3), 193–213 (2002)
14. Hu, B.G., Gosine, R.G., Cao, L.X., de Silva, C.W.: Application of a Fuzzy Classification Technique in Computer Grading of Fish products. IEEE Transactions on Fuzzy Systems 6(1), 144–152 (1998)
15. Zion, B., Shklyar, A., Karplus, I.: Sorting Fish by Computer Vision. Computers and Electronics in Agriculture 23(3), 175–187 (1999)
16. Fafioye, O.O., Oluajo, O.A.: Length-weight relationships of five fish species in Epe lagoon, Nigeria. African Journal of Biotechnology 4(7), 749–751 (2005)

Transformation of Products between Various Versions of the Rule World of a Product Configurator

Michael Fichter[1], Michael Klein[2], and Andreas Schmidt[1,3]

[1] Department of Computer Science and Business Information Systems,
University of Applied Sciences Karlsruhe, Germany
michael.fichter@web.de

[2] CAS Software AG, Karlsruhe, Germany
michael.klein@cas.de

[3] Institute of Applied Computer Science,
Forschungszentrum Karlsruhe, Germany
andreas.schmidt@kit.edu

Abstract. Product configurators allow for the generation/administration of valid product variants. For this purpose, a set of rules is filed in the configurator, which describe possible products that consist of a number of components and elements. If the underlying rule world of the configurator changes, for instance, due to a new legislation or the availability of components, etc., product variants that have been valid so far may become invalid. The present paper focuses on an approach to transforming products that have become invalid due to configuration changes back into valid products that are in accordance with the new rule basis. For this purpose, the concept of the *fictitious product* will be introduced and a two-stage transformation process proposed: In a first step, an invalid product is transformed into a valid fictitious product that is then converted into a valid real product.

1 Introduction

Today, products offered are becoming increasingly complex. On the other hand, the customer tends to demand an increasing individualization of the product properties. The times of Henry Ford saying: "Any customer can have a car painted any color that he wants so long as it is black" are long ago [3]. The span between individual single fabrication (expensive) and mass production (inexpensive) may be covered by so-called *mass customization.* Mass customization is characterized by a product being set up of a number of different, standardized assemblies or elements. The degree of individualization desired by the customer is achieved by selecting assemblies and elements appropriate for the customer.

In general, however, it is impossible to combine any assemblies or elements to an end product. Combinations may be limited. In case of a car, for instance, it is obvious that the installation of a more powerful engine may also require better brakes. A bicycle that is planned to have disk brakes also needs an appropriate frame to fix the brakes.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 721–730, 2009.

To support the setup of buildable products from individual components, so-called product configurators are used. They allow for reasonable combinations of individual components only and propose changes to the customer, if necessary. For this purpose, rules are formulated in the product configurator. They describe dependencies between the elements that can be used for construction. Additional limitations, such as cardinalities, are possible. All conditions and rules shall hereinafter be summarized by the term rule world. A product is *valid*, if the elements installed fulfill all conditions of the current rule world. Otherwise, the product is *invalid*.

In the course of time, the rule world of the product configurator may change. This may be due to the fact that a certain component is no longer available or to the availability of an improved successor or of cheaper alternative components or due to a changed legislation.

In this case, stored configurations of products/product variants may become invalid in the new rule world. It is now aimed at transforming products that have become invalid due to changes of rules into valid products (according to the new rule world). This transformation shall take place automatically, if possible. An invalid product P shall be transformed into a new product P', such that P and P' are as similar as possible.

The present paper will describe a multi-step approach to executing this transformation. The following section will concentrate on describing the functioning of a product configurator. Then, the transformation method will be presented and illustrated by means of an example. The paper will be completed by an outlook on future work relating to the transformation method and the underlying configurator.

2 Product Configurator

Product configurators belong to the category of expert systems [4]. The database represents the knowledge of the configurator. It consists of components, products, and limitations for the combination of components in the form of rules. Apart from the database, proper solution methods are applied to check configurations for validity based on the knowledge available and to calculate alternative solutions [5].

2.1 Functionality

The basic functionality of a configurator allows for transforming a valid product P together with a change desired into a new valid product P'. In P', all changes desired are met and all additionally needed reconstructions are made. If several possibilities of reconstructions exist, these are returned as alternatives. It is important to mention that the configurator calculates with valid products exclusively.

A product configurator is represented schematically in figure 1. Here, a model of the product configurator *CAS Configurator Merlin* [1] is shown.

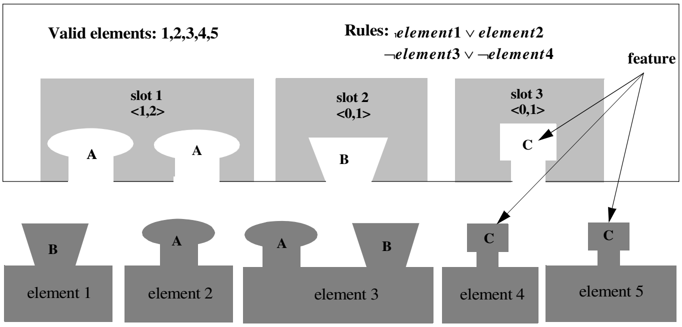


Fig. 1. Product Plan

2.2 Data Model

Each *product* is based on a so-called *product plan* that defines the equipment a *product* has to or may contain. *The product* plan represents all possible variants of a *product* . It is defined which *elements* can be used for these product variants. An example of an *element* is the *Winter Tire Polar 500*. *Element features* are properties implemented by *elements*. Hence, one or several *features* (in the example A, B, and C) are allocated to each *element*. In the example, the winter tires may implement the feature *tire*.

In addition, a number of *slots* are allocated to each product plan. A slot represents a point of installation of the elements. The minimum and the maximum cardinality are defined for each slot. This means the minimum and the maximum number of elements that may be installed in this slot. Each slot has exactly one feature to define which elements are permitted. Only elements implementing this feature may be installed in the slot. For example, the product plan might have the slot *car wheels* with the feature *tire*. Hence, the *winter tire Polar 500* (implements the feature *tire*) may be taken up. An element requires one free space in a slot at least. The element may also need several slots (element 3 in figure 1). An element installed in a slot is represented by a *slot filling*. Hence, a concrete product results from a number of slot fillings (the configuration). According to another concept, a product plan contains a set of rules. A rule represents the relations between the elements to be installed (e.g. if a car has an element *trailer hitch*, it must also have an element *bus bar* for additional backing lights). Any type of rule may be based on boolean operators. In figure 2, the data model is represented in the form of an ER model.

3 Conception of the Transformation Method

After having presented basic concepts of the product configurator in the previous section, the process to transform [2] an invalid product (according to the changed product plan) into a valid product shall be outlined now. In a first step, it is necessary to analyze which changes of a product plan can be made for a certain product to become invalid.

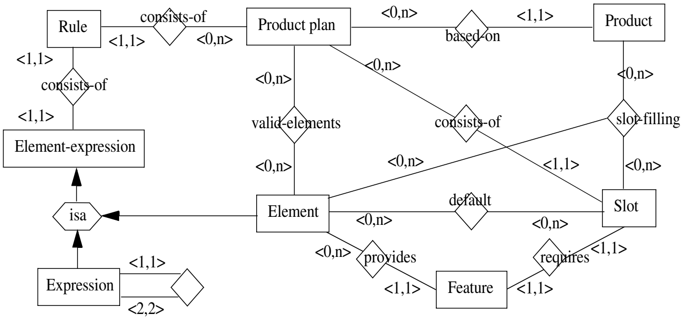


Fig. 2. ER model of the configurator

In detail, the following types of changes that may affect the validity of existing products can be distinguished:

1. Deletion of an element from the set of usable elements of a product plan: Products containing such an element become invalid.
2. Addition, change, or deletion of features of elements: When changing or deleting element features, the slot fillings (which represent the installation of the elements in a slot) do no longer contain any identical feature of slot and element. Products containing such changed elements become invalid.

   When features are added to elements, the original features of which have been installed in products, the added features are not yet installed in a slot. This also makes such products invalid (incomplete slot filling).
3. Deletion of a slot from a product plan: Products containing elements installed in such a slot become invalid.
4. Changing the feature of a slot: Products containing elements installed in such a slot become invalid, if the element feature installed in the slot does no longer agree with the slot feature.
5. Changing the cardinalities of a slot: More or less elements than permitted by the maximum and minimum cardinalities may be installed in a slot.
6. Adding rules to a product plan: As a result, the solution space of possible element combinations in a product is restricted and products may become invalid. The deletion of rules may not lead to invalid products, as a rule represents a restriction that is no longer valid when deleted.

Depending on the change of the product plan, the following errors of a concrete product may occur:

1. Invalid installed elements (changes type 1)
2. Invalid slot filling (changes types 2, 3, 4)
3. Violated cardinality (changes type 5)
4. Violated rule (changes type 6)
5. Incomplete slot filling (changes type 2)[1]

[1] Will no longer be considered with respect to product transformation.

3.1 Fictitious Elements and Products

To transform an invalid into a valid product, the concept of the *fictitious product* and *fictitious element* is introduced. A fictitious element is an element, for which dependencies on other elements in the form of rules are lacking (e.g. a fictive element *trailer hitch* does not need to have an element *bus bar* for additional backing lights). With the help of fictive elements rules can be avoided and thus invalid products can be corrected. Fictitious elements are supposed to represent element features and, hence, may replace concrete elements with the same features without violating rules. Accordingly, a *fictitious product* is any product that contains one fictitious element at least.

In the following sections, such fictitious elements shall be applied to correct errors that may be eliminated by the exchange or addition of elements.

3.2 Transformation Principle

In a first step, invalid products are transformed into valid fictitious products. All elements, the use of which leads to an invalid product, are removed or replaced by fictitious elements, such that a valid fictitious product results. In addition, new fictitious elements may be added.

In a second transformation, these valid fictitious products are transformed into valid real products. This is done by replacing the fictitious elements by really existing elements. This process is illustrated graphically by figure 3.

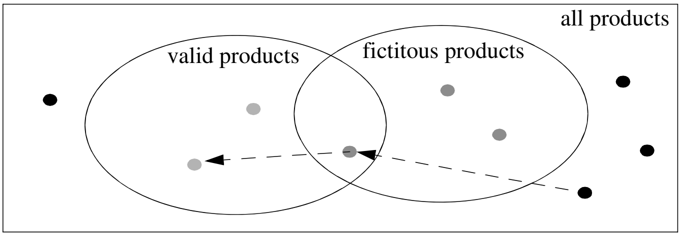


Fig. 3. Transformation principle

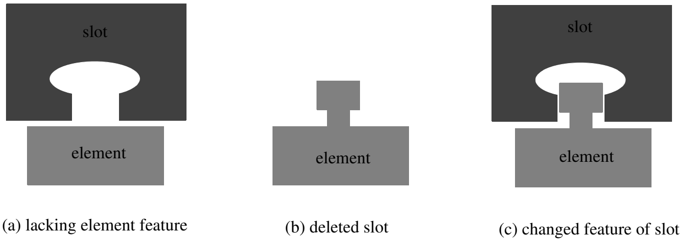
3.3 Algorithm

The complete transformation process is divided into five individual steps that shall be presented in detail.

The transformation steps are ordered such that steps which may lead to further errors are carried out prior to steps that correct them (see table 1). A secondary priority for the sequence of steps is the number of solution alternatives resulting after the execution of the step. Transformation steps resulting in a few solution alternatives are preferred to steps producing several alternatives (steps 4 and 5) in order to minimize the number of alternatives and calculation processes (see table 1).

Table 1. Prioritization of transformation steps

Step	Transformation step	Resulting errors	Increase in alternatives
1	Replacing invalid elements	None	No
2	Correcting invalid slot filling	Cardinality error	No
3	Eliminating minimum cardinality error	None	No
4	Eliminating maximum cardinality error	Rule error	Yes
5	Fulfilling rules	None	Yes



(a) lacking element feature (b) deleted slot (c) changed feature of slot

Fig. 4. Different invalid slot fillings

Step 1: In the first step, all invalid installed elements (error type 1) are replaced by fictitious elements having the same features as the invalid elements.

Step 2: Then, all invalid slot fillings (error type 2) are studied. Three cases can be distinguished(figure 4):

- lacking element feature (a)
- deleted slot (b)
- changed features of slot or element (c)

If no feature exists for an element, the element is removed from the product. This may cause minimum cardinality errors in the slot, which will have to be treated afterwards. If, however, the slot was deleted in the product plan, a new slot with this feature has to be searched for the element. This may give rise to maximum cardinality errors that will have to be treated afterwards. If no appropriate free slot is found, the element is removed from the product.

In the case of the changed feature, it is proceeded as in the previous case, and an appropriate slot is searched for the element (possibly, maximum cardinality error) or the element is removed from the product, if no such slot is found.

Step 3: Now, the cardinality errors are treated. Two cases are distinguished: (a) Minimum cardinality errors and (b) maximum cardinality errors. If the error results from remaining below the minimum cardinality, fictitious elements with the respective slot feature are added to the respective slots. This will not cause any further errors that will have to be treated later on.

In case of maximum cardinality errors, it has to be decided which of the elements have to be removed. Here, a prioritization is made and it is tried first to remove elements that violate rules. In this case, removal of the element has the positive side effect that a violation of the rule, which would have to be treated later on, is eliminated. Then, fictitious elements are removed, if the slot contains such fictitious elements as a result of one of the three previous transformation steps. Deletion of fictitious elements does not cause any further violations of rules, as fictitious elements do not possess any rule dependences on other elements. If the number of elements still is higher than the maximum cardinality, the next elements removed are elements, the deletion of which does not entail any further violations of rules. In the end, elements are removed, the deletion of which leads to violations of rules, which will have to be treated later on. If elements have to be deleted, the deletion of which results in violations of rules, solution alternatives are generated from the possible solution space. Then, these alternative solutions are analyzed. In the end, this may result in various valid products.

Step 4: Subsequently, the violated rules are analyzed. In a first step, the rules are transformed into the disjunctive normal form and it is distinguished between rules with exclusively negative terms (negative clauses), exclusively positive terms (positive clauses), and mixed clauses (both positive and negative terms).

Then, violations of rules are corrected in the following order depending on the individual types of clauses.

Negative clauses: To fulfill negative clauses, each element of the clause is replaced individually by a fictitious element. From the permutations, solution alternatives are generated for further processing.

Mixed clauses: They are fulfilled by replacing negative by fictitious elements (in analogy to negative clauses). Doing this, positive and other mixed clauses that contain the replaced element as a positive element may be violated. These mixed clauses are then corrected as well, until rules are not violated any longer. Positive clause errors are treated in the next step. To obtain a product that is as similar as possible to the original one, the information about the replacements by fictitious elements are stored and used in step 5 of the transformation process.

Positive clauses: This type of clause is corrected by installing each positive element individually and generating solution alternatives. When installing an element, further violations of rules may result and maximum cardinalities of slots may be exceeded (in the current version of a transformer, new violations of rules are not yet treated).

Step 5: In the last step, the fictitious elements are replaced by real elements. For this purpose, the real elements available with the same features are determined for each fictitious element. Then, replacement combinations are generated from the real elements available. Here, elements may be preferred, which are as similar as possible to the original elements of this feature.

As this may result in a large number of replacement combinations when the number of real elements available or the number of fictitious elements to be replaced is high, it is possible to reduce the replacement combinations to a certain number of alternatives. If the number of resulting alternatives exceeds a given threshold value, elements that do not violate any rules when installing are preferred to other elements with the same feature. This means that these elements are no longer used to calculate the replacement combinations, as a result of which the number of replacement combinations is reduced accordingly. The replacement combinations are installed individually for the fictitious elements in the fictitious product. The resulting products are checked for validity and the valid products are proposed as solution alternatives.

4 Example

Based on a concrete example, functioning of the transformation algorithm shall be demonstrated below. Table 2 shows the product example, table 3 the product plan underlying this product with all possible elements for the individual slots and cardinalities.

Now, a series of adaptations of the product plan are made. As a result, the rule world is changed:

1. The element *18" light metal* is removed from the product plan.
2. The minimum cardinality of the slot of equipment package is increased to 1.
3. The maximum cardinality of the slot of electronics is reduced to 1.
4. The rule $\neg Navigation\ device\ Path\ Finder \vee \neg Diesel\ engine$ is added.
5. The rule $\neg Diesel\ engine \vee Eskimo\ air\ conditioning$ is added.

Table 2. Product example

Slot	Element
Motorization	Diesel engine
Vehicle wheels	18" light metal
Electronics	Radio FM Deluxe
Electronics	Navigation device Pathfinder

Table 3. Orginal product plan

Feature	Slot	Possible elements
Drive	Motorization <1,1>	Diesel engine
		Otto engine
Tires	Vehicle wheels <1,1>	18" light metal
		19" light metal
Entertainment electronics	Electronics <0,n>	Radio FM Deluxe
		CD juggler
		Navigation device Pathfinder
Basic equipment	Equipment package <0,1>	Sports equipment
		Comfort equipment
Special equipment	Others <0,1>	Eskimo air conditioning

Table 4. Product after step 1

Slot	Element
Drive	$Motorization_{fictitious}$
Vehicle wheels	$Tires_{fictitious}$
Electronics	Radio FM Deluxe
Equipment package	$Equipment\ package_{fictitious}$

Table 5. Product variant 1 and 2 after step 5

Slot	Element
Drive	Otto engine
Vehicle wheels	19" light metal
Electronics	Radio FM Deluxe
Equipment package	Comfort equipment (or Sports equipement)

These changes lead to the following errors:

Type 1: Invalid element: *18" light metal wheels* installed
Type 2: Minimum cardinality violated: No equipment package defined
Type 3: Maximum cardinality violated: Both radio and navigation device installed
Type 4: Violation of rule:
$\neg Navigation\ device\ Pathfinder \vee \neg Diesel\ engine$
Violation of rule:
$\neg Diesel\ engine \vee Eskimo\ air\ conditioning$

To obtain a valid product according to the new product plan again, the following steps are executed depending on the error types:

- **Step 1:** The element *18" light metal* is replaced by a fictitious element with the feature *tires*.
- **Step 2:** No invalid slot fillings have to be fixed.
- **Step 3 (a):** To fulfill the minimum cardinality of the slot of equipment package, a fictitious element of the type *special equipment* is installed in the slot.
- **Step 3 (b):** The maximum cardinality of the slot *electronics* can be observed by removing either the radio *FM Deluxe* or the navigation device *Path-Finder*. As the navigation device *Path-Finder* also violates the rule $\neg Navigation\ system\ Pathfinder \vee \neg Diesel\ engine$ element is preferably removed from the product. No further rule errors result and maximum cardinality is observed.
- **Step 4:** The rule $\neg Diesel\ engine \vee Eskimo\ air\ conditioning$ remains as a rule error. According to the previously formulated rule for mixed clauses, it can be fulfilled by replacing the *Diesel engine* by a fictitious element with the feature *motorization*.

  These changes result in the following valid fictitious product in table 4:
- **Step 5:** After replacing the fictitious elements by real elements, two possible real products (see table 5) result.

5 Conclusion

This paper presents an approach to transforming products that have become invalid due to configuration changes into valid products again. For this purpose, an transformation method was developed, by means of which one or several proposals of valid products are submitted for invalid products via an auxiliary fictitious element. For this purpose, various error classes are identified and various replacement/exchange and deletion strategies are developed.

The method still is incomplete, as no transformation into a valid product is possible for the error class of incomplete slot filling. The same applies to rule evaluation when successive errors result from correcting positive clauses. These two issues are subject of current research work.

It is considered to extend the product configurator by properties for elements. It would be possible to provide certain elements with additional properties (e.g. battery, property: *capacity = 60 ampere hours*). Hence, it would not only be possible to enforce or exclude the existence of a certain element in rules, but to extend the rules to cover the formulation of properties. A possible rule of a computer may be the following: If a car possesses more than two elements in electronic slots, the battery must provide 60 ampere hours at least.

References

1. CAS: An overview on CAS Configurator Merlin functions, http://www.cas.de/merlin
2. Fichter, M.: Transformation von Produkten zwischen verschiedenen Versionen der Regelwelt eines Produktkonfigurators: Untersuchung und prototypische Implementierung Bachelor Thesis University of Applied Sciences, Karlsruhe (2008)
3. Ford, H., Crowther, S.: My life and work Doubleday Erewhon, NC (1922)
4. Grasmann, M.: Produktkonfiguratoren auf Basis von Engineering-Data-Management-Systemen Dissertation, University of Paderborn (2000)
5. Günter, A., Kühn, C.: Knowledge-based configuration- survey and future directions. In: Puppe, F. (ed.) XPS 1999. LNCS, vol. 1570, pp. 47–66. Springer, Heidelberg (1999)

Constraint Satisfaction Approach on Product Configuration with Cost Estimation

Lin Wang[1], Wee Keong Ng[1], and Bing Song[2]

[1]School of Computer Engineering, Nanyang Technological University of Singapore,
Singapore 639798
[2]Singapore Institute of Manufacturing Technology, Singapore 638075
wang0328@ntu.edu.sg, awkng@ntu.edu.sg, bsong@SIMTech.a-star.edu.sg

Abstract. Due to its flexibility and generality, the constraint satisfaction problem (CSP) has been considered as an effective approach for solving product configuration problems. However, classical CSP-based configuration systems are not powerful enough to capture the inherent natures of configuration problems, such as undetermined number of components and the internal structure of components. We propose an extended constraint satisfaction schema that captures and takes advantage of the hierarchy of constituents and the derivation relationships among those constituents from the view of product family design, such that the constraint network of the CSP-based configuration extends incrementally. We also integrate cost estimation with the configuration process.

Keywords: Product configuration, Constraint satisfaction problem, Product family, Cost estimation, Mass customization.

1 Introduction

In the last two decades, mass customization has been the trend in manufacturing to satisfy diverse customers' requirements while keeping near mass production efficiency [1,2]. Product family design is widely accepted as the most effective approach for mass customization [5,6]. Product configuration is the task of selecting and arranging combinations of pre-defined components based on a set of well-defined combination rules in order to satisfy given requirements for solving product family design problems [13].

Many efforts have been made in modeling product family [6,7] and developing applicable approaches to assist product configuration; e.g., case-based configuration [8], rule-based configuration [9], and resource-based paradigm [10]. Highly declarative, domain independent and simple to use, Constraint Satisfaction Problem (CSP) is a promising approach for modeling and solving product configuration problems. CSP can be simply described as a problem composed of a set of variables, each of which is associated with a domain, and a set of constraints that restrict the combination of values that the variables can simultaneously take. The task of a CSP is to find solutions that assign a value to each

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 731–740, 2009.

variable satisfying all the constraints. Product configuration problems can easily be modeled as a CSP with the concept of variables, domains, and constraints. Thereafter, the CSP can be solved by various solving algorithms independent of the configuration knowledge. However, the basic formalism of CSP is not comprehensive enough to represent complex configuration problems where the number of components is not known a *priori*, and fails to capture the inherent hierarchical structure of components in product configuration problems.

In mass customization environment, cost is as important as functionality and quality in the terms of the success of a product [3]. Proper cost estimation before final product configuration is critical for a competitive product. As product design influences about 70%-85% of the total cost of a product [4], separating the product configuration process from cost estimation reduces the effectiveness of both tasks. By integrating proper cost estimation into product configuration, a product is designed with good combination of optimal functionality, quality, and cost.

In this paper, we propose an extended dynamic CSP-based configuration model that incorporates structural constraints and cost constraints. Using this model, product configuration and cost estimation are represented as a CSP. This CSP model makes use of the activation and structural constraint to extend a CSP-based configuration schema that allows incremental extension of a constraint network while capturing the internal hierarchy of components in the product family design. We also provide a formalism to express cost relationships in CSP terms and integrate cost estimation into the product configuration processes.

The paper is further organized as follows: Section 2 is the literature review on product configuration. We propose in detail our extended DCSP-based configuration model with cost estimation in Section 3. Finally, conclusions are drawn in Section 4.

2 Literature Review

Product configuration is the task of selecting and arranging combination of appropriate sets of components to satisfy given customers' requirements. Several approaches have been developed to model and solve configuration problems.

Case-based configuration solves a new configuration task by comparing new requirements with historical data to find similar requirements, based on which similar solutions are retrieved [8]. Minor adaptation is needed to refine similar solutions to match the new requirements. This approach is easy and does not need any knowledge acquisition phase. However, it may not generate a complete solution as certain knowledge is difficult to retrieve from historical cases. So it is not suitable for any configuration with new techniques. It may result in inaccuracy when the size of historical data is not sufficiently large.

Rule-based approach uses production rules of the form "*if* condition *then* consequence" to represent both control and domain knowledge [9]. The if-then structure makes knowledge representation easy and intuitive. However, as configuration systems grow, the rule base becomes very large. This method lacks the separation of control and domain knowledge. All of these may incur huge maintenance overhead, which is difficult to eliminate [13].

Resource-based method treats a configuration as the supply and consumption of resources among elements including components and the environment [10]. It offers a producer-consumer model in which each element is characterized by the amount and type of resources it supplies and consumes. The final solution is derived when the overall resources reach a balance state. It offers an advantage in the case of function-component matching. However, this method cannot solve the component structure problem, which is an important issue in configuration tasks. In addition, it loses its efficiency one considers technical constraints, which are necessary for most of technical configuration systems.

Constraint satisfaction problem (CSP) has proven to be a promising approach to model and solve configuration problems. The first attempt to apply CSP to configuration problem was presented by Mittal and Frayman (1989) [11]. In their approach, a product is composed of a fixed, pre-defined set of components. Each component possesses a set of properties and ports for connecting to other components, which are represented as variables with discrete and finite domains. The restrictions on how to organize the components to form a valid configuration are represented as compatibility constraints. Thus, a product configuration modeled as a CSP can be achieved by assigning a value to each variable without violating any constraints. CSP-based product configuration has the advantage of flexibility and generality [15]. However, due to the assumption of a fixed and predefined set of components, the classical approach is limited in expressiveness for more complex configuration. For instance, the set of variables of a solution cannot be predetermined. It also fails to capture the inherent structure of components of a product.

To address the first issue of classical CSP-based configuration approach, Mittal and Falkenhainer presented dynamic constraint satisfaction (DCSP) based configuration system [12], which is an extension of the classical approach. In DCSP-based configuration, only a subset of all the variables need to be instantiated and eventually adopted in the solution, depending on selections. Each variable holds one of two states: active or inactive. Only active variables are involved in the solution. Besides compatibility constraints, activity constraints are introduced to specify conditions on the values of active variables, under which inactive variables become active. This approach is more powerful; however, it still suffers from the limitation of unclear component structure.

3 DCSP-Based Configuration System with Cost Estimation

As we mentioned previously, product family is an effective approach towards mass customization and cost estimation should be integrated into the product configuration process. In this paper, we propose an extended DCSP-based configuration system to encompass the concept of product family and cost estimation. We also extend the DCSP system with structural constraints to capture internal hierarchical information in a product configuration task. Using this system, a product configuration problem can be described as an incrementally extended constraint network, together with relevant cost information and the internal hierarchy of modules/components in product family design.

3.1 Configuration Domain

The DCSP-based configuration includes or excludes variables involved in the final solution dynamically. The active variables and associated active constraints are derived from a predefined knowledge base called the configuration domain.

Definition 1. ***Configuration Domain.*** *A configuration domain Φ is described as $\langle V_M, V_{CM}, V_P, V_{CT}, D, C\rangle$, where*

- *V_M: A possible infinite set of module variables that are the main constitutions of a product family.*
- *V_{CM}: A possible infinite set of component variables that are the instances of corresponding module variables. The domain of each variable from V_M is a finite set of component variables; i.e., Domain(V_{M_i}) $\subseteq V_{CM}$.*
- *V_P: A set of property variables that are associated with modules and components. The domain of each variable from V_P contains numeric or symbolic values that can be assigned to the property variable.*
- *V_{CT}: A set of cost variables that are associated with modules and components. The domain of each cost variable includes non-negative numeric values.*
- *D: The set of domains. Each domain is a non-empty set containing all possible values for the corresponding variables.*
- *C: A finite set of constraints, including activation constraints, structural constraints, cost constraints, etc.*

In this definition, modules of a product family, components of product variants, properties and cost drivers associated with the modules and components are all represented as variables in the CSP model. For product configuration, modules are specified in a product family architecture (PFA). According to customer specifications, components of a product corresponding to the modules in the product family are selected, and consequently, properties and cost drivers associated with selected modules and components are activated by *activation constraints.*

The name of property variables is composed from the name of the originating module or component and the property name; e.g., if property $p \in V_P$ is defined for module $m \in V_M$, then the property variable is *m.p*. Similarly, property $p \in V_P$ for component $c \in V_{CM}$ is named *c.p*. The set of all possible property variables can thus be specified as $V_P = \{m.p \cup c.p \mid m \in V_M,\ c \in V_{CM},\ p \in V_P\}$. Similarly, the set of all possible cost variables can be specified as $V_{CT} = \{m.t \cup c.t \mid m \in V_M,\ c \in V_{CM},\ t \in V_{CT}\}$. Note that the properties and cost variables for a module are all inherited by instance components.

3.2 Constraints

In CSP-based product configuration and cost estimation, constraints are the driver of configuration and estimation processes. Constraints stipulate the rules that product configuration follows and restrict the combination of values that the variables can simultaneously take. In our DCSP model, constraints consist of activation constraints, structural constraints, cost constraints, etc. In this section, we define these constraints respectively.

Activation Constraints. In product configuration, property variables and cost variables belong to specific modules or components. Thus, when a module variable is selected in a DCSP-based configuration system, one of its instance component variables is chosen from the its domain to satisfy specific requirements and the appropriate sets of property variables and cost variables associated with the selected module and component are introduced into the DCSP system. Such derivation relationships between the variables in the DCSP model are expressed by *activation constraints* in the general form of

$$var_1 \Longrightarrow var_2 \ , \tag{1}$$

where $var_1 \in V_M \cup V_{CM}$ and $var_2 \in V_P \cup V_{CT}$.

To avoid the case where the activation of var_2 is not triggered by var_1, we also assume that in a configuration solution, if var_2 is active, then var_1 must be active. For example, if a property variable $c.p$ for component c is active, then component c must be active and its corresponding module m must also be active.

In configuration domain $\Phi = \langle V_M, V_{CM}, V_P, V_{CT}, D, C \rangle$, $m \in V_M$ is a module selected in a PFA, $c \in V_{CM}$ is a component corresponding to m, $p \in V_P$ is a property of c, $t \in V_{CT}$ is a cost variable of c. The derivation relationships in product configuration and cost estimation task are represented by the following activation constraints:

$$m \implies m.p, \ m \implies m.t, \ c \implies c.p, \text{ and } c \implies c.t \ . \tag{2}$$

By activation constraints, the properties and costs variables are added to the active variables of a CSP model dynamically.

Structural Constraints. The hierarchical nature of modules/components is an important content in product configuration tasks. In product family based configuration, the structure of product variants are derived from the corresponding product family. Thus, the hierarchy of components in a product can be viewed as an instance of the hierarchy of modules in the corresponding product family. In our extended DCSP-based model, we make use of structural constraints to capture the hierarchical information among modules. Consequently, structural constraints can be applied to the corresponding components.

In PFA, a product family configuration can be represented as a module tree. A module is either a root module, or a internal module, or a leaf module. The root module and internal modules have at least one direct child module. Then in configuration domain $\Phi = \langle V_M, V_{CM}, V_P, V_{CT}, D, C \rangle$, $m_1 \in V_M$ is a non-leaf module in a PFA, $m_{11}, m_{12}, \ldots, m_{1t} \in V_M$ are a set of direct child modules of m_1. Such a structure can be expressed by the following structural constraint:

$$m_1 \Downarrow \{m_{11}, m_{12}, \ldots, m_{1t}\} \ . \tag{3}$$

Cost Constraints. To integrate cost estimation with the configuration task tightly, we augment our DCSP-based configuration system by introducing cost constraints. These constraints use cost functions in arithmetic constraint expressions.

Definition 2. ***Cost Function.*** *A cost function τ is a tuple $\langle \varphi, \delta \rangle$, where φ is a cost calculation function (such as $\sum$, $\prod$, sigmoid, max, min) and $\delta \subseteq V_{CT}$ is a set of cost variables. The cost function τ is computed by applying φ to the set $\{m.t \cup c.t \mid (m.t \cup c.t) \in \delta \wedge active(m.t \cup c.t)\}$.*

In the cost function, the cost variables involved in δ must be active. The cost function on the cost variables of modules, $m.t$, are inherited by their corresponding components. A constraint that contains a literal with a cost function is called a *cost constraint.* For example:

$$Cost_{\text{resource}} = \sum_{i=1}^{n} CR_i \times Qt_i \ , \tag{4}$$

where $Cost_{\text{resource}}$ is the total resource cost consumed by a activity; n is the number of different resources consumed by the activity; CR_i is the consumption rate of the i_{th} resource; and Qt_i is the consumption quantity of the i_{th} resource. The right side of the cost constraint is a cost function, where $\sum$ is the cost calculation function, CR_i and Qt_i are the cost variables.

For implementation purposes, we distinguish two categories of cost constraints: Cost calculation constraints and cost variation constraints. Cost calculation constraints are used to calculate the cost, or state the bound of the cost of specific components or activities in the configuration process. The example above is just a cost calculation constraint.

In the product family based product configuration, impact analysis of the change of design parameters or cost drivers on the product cost is also an important issue. A small change of a design parameter or a cost driver may positively or negatively induce the change of numerous cost drivers, and finally the change of the total product life cycle cost may be huge. Cost variation constraints are created to capture the cost impact relationships among cost drivers.

3.3 Configuration Problems

A product configuration problem can be described as a DCSP that consists of a configuration domain referring to a knowledge base comprising various variables and the constraints governing them, a set of initial variables with corresponding domains, and a set of constraints related to initial variables.

Definition 3. ***Configuration Problem.*** *A configuration problem is a quaternion $\langle V^I, D^I, C^I, \Phi\rangle$, where $\Phi = \langle V_M, V_{CM}, V_P, V_{CT}, D, C\rangle$ is a configuration domain, $V^I \subseteq V_M \cup V_{CM} \cup V_P \cup V_{CT}$ is the set of initial variables, $D^I \subseteq D$ is the corresponding domains of V^I, and $C^I \subseteq C$ is the set of constraints on V^I. A solution to a configuration problem is an assignment to a set of activated variables V^a ($V^a \subseteq V_M \cup V_{CM} \cup V_P \cup V_{CT}$, $V^I \subseteq V^a$), such that all constraints on V^a are satisfied.*

The variables in V^I with corresponding D^I and the constraints in C^I compose a partial constraint network as the initial status of a DCSP-based product configuration task, and then extend dynamically to achieve the solution of the DCSP-based configuration problem.

4 Solving DCSP-Based Configuration Problem

Based on the DCSP-based configuration problem formalized previously, the solving process of a configuration problem $\langle V^I, D^I, C^I, \Phi\rangle$ dynamically constructs a constraint network N starting from an initial network $\langle V^I, D^I, C^I\rangle$. The network will extend for the following reasons:

- When a module m_1 is selected, all its child modules, which are not existing modules in N, are added to the active variables of N as required by the structural constraints of the form $m_1 \Downarrow \{m_{11}, m_{12}, \ldots, m_{1t}\}$.
- When a module m is selected for an assignment, an instance component c of m chosen from m's domain is introduced to N as an active variable.
- When a module m and its corresponding component c are selected, all the property variables and cost variables on m and c are introduced to N as the active variables by activation constraints of the form m $\Longrightarrow$ m.p, c $\Longrightarrow$ c.p, m $\Longrightarrow$ m.t, or c $\Longrightarrow$ c.t.
- Further, the constraints whose variables are all the active variables are added to N as active constraints.

The solving process of our DCSP-based configuration system can be viewed as a sequence of stable phases. During each stable phase, the constraint network N can be viewed as a fixed classical constraint network, which only includes currently activated variables and associated activated constraints. A new phase is started by the activation of new variables with the subsequent introduction of new active constraints. In each stable phase, CSP solving algorithms are applied such that constraints on the active variables are propagated until a consistent assignment to all the active variables is reached.

Among the various solving algorithms of CSP, the backtracking paradigm is the most popular method, where variables are instantiated sequentially, and the validity of the constraint is checked once all its variables are instantiated. Backtracking is performed to reevaluate the most recent instantiated variable when some instantiation of variable violates the constraints. Arc consistency checking is implemented to enhance the efficiency and eliminated the thrashing

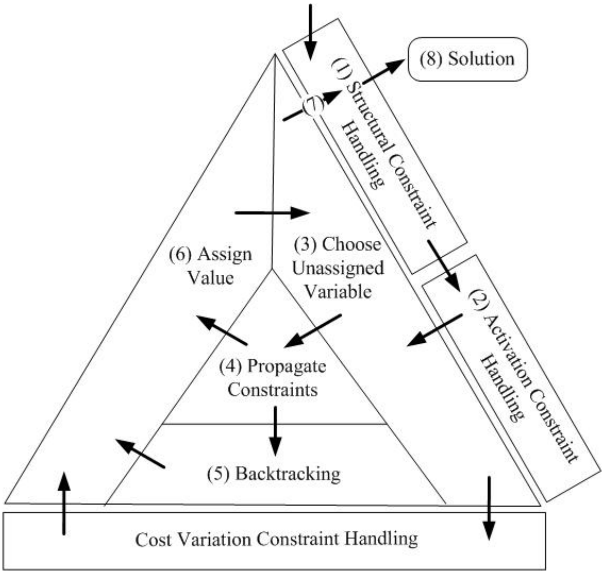

(8) Solution
(1) Structural Constraint Handling
(7)
(6) Assign Value
(3) Choose Unassigned Variable
(2) Activation Constraint Handling
(4) Propagate Constraints
(5) Backtracking
Cost Variation Constraint Handling

Fig. 1. DCSP-based configuration solving process

caused by backtracking paradigm. A backtracking search process is illustrated in Fig. 1, where structural constraint handler and cost variation constraint handler are embedded into a classical searching procedure for discrete DCSPs. The main loop of the process consists of the following steps:

- Step 1: In the structural constraint handling phase, when a module variable is added into the constraint network N, its direct child module variables are activated and added into N. A component variable corresponding to each module variable is also added into N.
- Step 2: By activation constraints, new property variables and cost variables associated with the activated module and component variables are generated and introduced into N. By now, a new stable constraint network is generated.
- Step 3: In the stable constraint network N, an unassigned variable is chosen as the current variable to check.
- Step 4: Assuming an unassigned variable exists, constraint propagation is performed on N, such that inconsistent values are (temporally) removed from the current domains of unassigned variables; i.e., all values of the current variable are locally consistent with the assigned variables.
- Step 5: If a variable domain becomes empty, backtracking is performed.
- Step 6: A value is assigned to the current variable from its (remaining) domain, which is consistent to the partial assignment. Steps 3-6 iteratively perform to assign a value to each currently active variables in N.
- Step 7: In Step 3, if no unassigned variable exists, the process returns to Step 1. New child module variables for an existed module variable, together with corresponding component variables and associated active constraints,

are generated and added into *N*. New searching phase is formed and solving process continues.

- Step 8: If no new child module variable exists, the constraint solving process terminates and a solution of the CSP is found.

The solving process presented above generates a new stable constraint network and proceeds the general backtracking process on it. If a consistent assignment to all the active variables is reached without new variables being activated, the assignment is a solution. If new module variables and component variables are generated, the solving process continues with the new extended network.

In this solving process, two categories of cost constraints are implemented in different phases. In the constraint propagate phase (step 4), cost calculation constraints act as ordinary constraints checking consistency of the constraint network *N*, and calculate the result of cost estimation. Thus when a configuration solution is reached, the cost estimation solution is achieved simultaneously.

In general product configuration and cost estimation processes, cost variation constraints are not activated and introduced to the constraint network *N*. They are implemented in the cost variation constraints handling phase to estimate the cost impact of any change of variables on the total product (family) cost. With a specific product configuration, we choose a changed variable as the current variable to check (step 3). By cost variation constraints, directly affected variables are added to the constraint network, and the changed-value of those variables are calculated and assigned to them. This process continues iteratively until all the affected variables are identified and introduced into the constraint network. Finally, by cost calculation constraints, the cost impact result can be achieved. Taken in this sense, cost variance constraints trigger new active variables as the activation constraints to extend the constraint network.

5 Conclusions

We have proposed a coherent formalism for modeling and solving product family based configuration problems integrating with the cost estimation task. This schema extends classical CSP-based configuration by identifying the derivation of the existence of variables in product configuration and cost estimation problems. This formalism is therefore suitable for modeling problems where the number of variables involved in a solution is not known a *priori*. We have identified three specific types of constraints: Structural constraints, activation constraints and cost variation constraints, that may lead to the dynamic activation of new variables.

In this schema, the product configuration is derived from the level of product family. We have extended the classical CSP-based configuration by structural constraints to capture the inherent internal hierarchy of modules/components in product configuration problems. This hierarchical nature is also used to trigger new active module variables into the configuration solving process. To integrate cost estimation with product configuration, we have proposed cost constraints represented by mathematical formalisms to estimate the product cost and cost impact of variable changes. Finally, a backtracking solving process has also been proposed and the influence of the variables presented above on the operation of the solving process is also discussed.

References

1. Boothroyd, G., Dewhurst, P., Knight, W.A.: Product Design for Manufacture and Assembly. Marcel Dekker, New York (1994)
2. Fixson, S.K.: Assessing Production Architecture Costing: Production Life Cycles, Allocation Rules, and Cost Models. In: ASME 2004 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Salt Lake City (2004)
3. Layer, A., Brinke, E.T., Houten, F.V., Kals, H., Haasis, S.: Recent and Future Trends in Cost Estimation. INT. J. Computer Integerated Manufacturing 15(6), 499–510 (2002)
4. Niazi, A., Dail, J.S., Balabani, S., Seneviratne, L.: Product Cost Estimation: Technique Classification and Methodology Review. Journal of Manufacturing Science and Engineering 128, 563–575 (2006)
5. Davis, S.: From Future Perfect: Mass Customization. J. Planning Review 17(2), 16–21 (1989)
6. Du, X.H., Jiao, J.X., Tseng, M.M.: Architecture of Product Family: Fundamental and Methodology. J. Concurrent Engineering: Research and Application, 9(4), 309–325 (2001)
7. Jiao, J.X., Tseng, M.M.: Fundamentals of Product Family Architecture. Journal of Engineering Design 11(3), 225–243 (2000)
8. Heylighen, A., Neuckermans, H.: A Case Base of Case-Based Design Tools for Architecture. J. Computer Aided Design 33(14), 1111–1122 (2001)
9. McDermott, J.: R1: A Rule-Based Configurer of Computer Systems. J. Artificial Intelligence 19(1), 39–88 (1982)
10. Heinrich, M., Jungst, E.W.: A Resource-Based Paradigm for the Configuring of Technical Systems from Modular Components. In: 7th IEEE Conference on Artificial Intelligence Applications, pp. 257–264. IEEE Press, Miami (1991)
11. Mittal, S., Frayman, F.: Towards a Generic Model of Configuration Tasks. In: Proc. 11th Int. Joint Conf. of Artificial Intelligence, Detroit, pp. 1395–1401 (1989)
12. Mittal, S., Falkenhainer, B.: Dynamic constraint satisfaction problems. In: Proceedings of the 8th National Conference on Artificial Intelligence, pp. 25–32. AAAI Press, Boston (1990)
13. Sabin, D., Freuder, E.C.: Configuration as Composite Constraint Satisfaction. In: Proceedings of the Artificial Intelligence and Manufacturing Research Planning Workshop, pp. 153–161. AAAI Press, Albuquerque (1996)
14. Stumptner, M., Haselbock, A.: A Generative Constraint Formalism for Configuration Problems. In: 3rd Congress of Italian Association for Artificial intelligence, Torino, Italy, pp. 302–313 (1993)
15. Xie, H., Henderson, P., Kernahan, M.: Modelling and Solving Engineering Product Configuration Problems by Constraint Satisfaction. International Journal of Production Research 43, 4455–4469 (2005)

Compression-Based Measures for Mining Interesting Rules

Einoshin Suzuki

Kyushu University, Fukuoka 819-0395, Japan
suzuki@i.kyushu-u.ac.jp
http://www.i.kyushu-u.ac.jp/~suzuki

Abstract. An interestingness measure estimates the degree of interestingness of a discovered pattern and has been actively studied in the past two decades. Several pitfalls should be avoided in the study such as a use of many parameters and a lack of systematic evaluation in the presence of noise. Compression-based measures have advantages in this respect as they are typically parameter-free and robust to noise. In this paper, we present J-measure and a measure based on an extension of the Minimum Description Length Principle (MDLP) as compression-based measures for mining interesting rules.

Keywords: Interestingness measure, Rule Discovery, Minimum Description Length Principle, J-measure, Data Mining.

1 Introduction

In the last two decades, measures for mining interestingness rules have been actively studied e.g. [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22]. An interesting rule goes beyond an accurate rule as the former is related with many factors including the accuracy. Interestingness measures for rules can be classified as either objective or subjective depending on whether the measure uses only discovered patterns and the data from which the patterns are discovered, or the measure uses additional information such as domain knowledge [23].

Defining human's interestingness can be called as AI-hard as it is as difficult as all problems in artificial intelligence (AI) [23]. What is interesting depends on various factors including the task, the individual, and the context; and any problem and any combination of problems in AI can be used to invent a challenging situation for interestingness measures for rules. We must beware of the hype of omnipotent measures and we must settle realistic objectives for research on measures for mining interesting rules.

Desiderata on interestingness measures for rules can be classified into qualitative expressions such as generality, accuracy, simplicity, and comprehensibility; and quantitative relations such as [12,21]. We have pointed out desirable properties in exception rule/group discovery such as interpretation of the evaluation measure, which supports the quality of discovered patterns [24]. Pitfalls to avoid are much less known than the desiderata and include four biases in evaluation

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 741–746, 2009.

[18] and a use of many parameters [25], the latter of which poses extra work on the users and results in a problem that is analogous to overfitting in classification.

Compression-based methods have been successful in classification as they are robust against noise and have theoretical bases. In this paper, we introduce J-measure of ITRULE [15] and a measure based on an extension of the MDLP of CLARDEM [19] both of which belong to the compression-based approach and thus exhibit attractive properties including parameter-freeness.

2 Rule Discovery

The objective of rule discovery is to obtain a set Π of rules given data D and additional information α. Here α typically represents domain-specific criteria such as an expected profit or domain knowledge, and an element of Π represents a rule π. In the objective approach $\alpha = \emptyset$ while in the subjective approach $\alpha \neq \emptyset$.

The case in which D represents a table, alternatively stated "flat" data, is most extensively studied in data mining. On the other hand, the case when data D represent structured data such as time-series data and text data typically necessitates a procedure for handling such a structure (e.g. [26,27]). In order to focus on the interestingness aspect, we limit our attention to the former case. In this case, D consists of n examples $e_1, e_2, \ldots, e_n$. An example e_i is described with m attributes $a_1, a_2, \ldots, a_m$ and an attribute a_j takes one of $|a_j|$ values $v_{j,1}, v_{j,2}, \ldots, v_{j,|a_j|}$. We represent $e_k = (w_{k,1}, w_{k,2}, \ldots, w_{k,m})$, where $w_{k,j} \in \{v_{j,1}, v_{j,2}, \ldots, v_{j,|a_j|}\}$.

A rule represents a local probabilistic tendency in D and can be represented as $A \rightarrow B$. Here A and B are called a premise and a conclusion, respectively, and each of them specifies a subspace of the example space. For instance, $(a_1 = v_{1,1}) \vee (a_2 = v_{2,2}) \rightarrow (a_3 = v_{3,1}) \wedge (a_4 = v_{4,4})$ is a rule. A rule can be classified into either logical or probabilistic and most of the papers in data mining are concerned with the latter. A probabilistic rule $A \rightarrow B$ can have a confidence $P(B|A)$ smaller than 1 i.e. $P(B|A) < 1$ while a logical rule necessitates $P(B|A) = 1$. Note that here we use each of A and B to represent a set of examples which reside in the corresponding subspaces.

3 Compression-Based Measures for Mining Interesting Rules

3.1 J-Measure

In computer science, research on objective interestingness measures for rule discovery goes at least back to 1960's [28,29]. Various measures including [6] have been proposed. Since the objective of this paper is not to conduct an extensive survey on such measures, we first explain the J-measure [15] as a representative of the compression-based measures.

We call an assignment $a = v$ of a value v to an attribute a an atom. A literal is defined as either a single atom or a conjunction of multiple atoms.

Consider a rule $\boldsymbol{Y} = \boldsymbol{y} \rightarrow X = x$, where the premise $\boldsymbol{Y} = \boldsymbol{y}$ is a literal and the conclusion $X = x$ is an atom[1]. Here $\boldsymbol{Y}$ and $\boldsymbol{y}$ correspond to a vector of attributes and a vector of attribute values, respectively. We define $f(X, \boldsymbol{Y} = \boldsymbol{y})$ as the instantaneous information that the event $\boldsymbol{Y} = \boldsymbol{y}$ provides about X, i.e. the information that we receive about X given that $\boldsymbol{Y} = \boldsymbol{y}$ has occurred. It is shown in [28] that the following j-measure is the only non-negative function that satisfies $E_{\boldsymbol{y}}[f(X; \boldsymbol{Y} = \boldsymbol{y})] = I(X; \boldsymbol{Y})$, where $E_{\boldsymbol{y}}(g)$ and $I(X; \boldsymbol{Y})$ represent the expected value of g in terms of $\boldsymbol{y}$ and the mutual information between X and $\boldsymbol{Y}$, respectively.

$$
\begin{aligned}
j(X; \boldsymbol{Y} = \boldsymbol{y}) &= \sum_x P(x|\boldsymbol{y}) \log_2 \left(\frac{P(x|\boldsymbol{y})}{P(x)} \right) && (1) \\
&= P(x|\boldsymbol{y}) \log_2 \left(\frac{P(x|\boldsymbol{y})}{P(x)} \right) + P(\overline{x}|\boldsymbol{y}) \log_2 \left(\frac{P(\overline{x}|\boldsymbol{y})}{P(\overline{x})} \right) && (2)
\end{aligned}
$$

(2) is derived due to the nature of rule discovery i.e. it suffices to consider the events of the conclusion $X = x$ and its negation $X = \overline{x}$. In [15], the interestingness of the rule $\boldsymbol{Y} = \boldsymbol{y} \rightarrow X = x$ is evaluated with the average information content of the rule under the name of the J-measure $J(X; \boldsymbol{Y} = \boldsymbol{y})$.

$$
J(X; \boldsymbol{Y} = \boldsymbol{y}) = P(\boldsymbol{y}) j(X; \boldsymbol{Y} = \boldsymbol{y}) \tag{3}
$$

Note that the J-measure represents the amount of information compressed by the rule $\boldsymbol{Y} = \boldsymbol{y} \rightarrow X = x$ with the rule, the code length for an example satisfying $x \wedge \boldsymbol{y}$ becomes $-\log_2 P(x|\boldsymbol{y})$ from $-\log_2 P(x)$ and the code length for an example satisfying $\overline{x} \wedge \boldsymbol{y}$ becomes $-\log_2 P(\overline{x}|\boldsymbol{y})$ from $-\log_2 P(\overline{x})$. (3) represents the difference of the amount of information between the situations with and without the rule. From the practical viewpoint, a general rule ($P(\boldsymbol{y})$ high), an accurate rule ($P(x|\boldsymbol{y})$ high), and a rule predicting a rare event ($P(x)$ low) exhibits a high J-measure value if other parameters remain unchanged. This characteristic makes sense from the viewpoint of discovery of interesting rules with an interestingness measure.

3.2 Measure Based on an Extended MDLP

We have proposed a subjective interestingness measure for groups of classification rules which are mutually related based on the Minimum Description Length Principle [19]. Unlike conventional methods, our interestingness measure is at the same time based on a theoretical background, has no parameter, is applicable to a group of any number of rules, and can exploit an initial hypothesis.

An example $(v_{i,1}, v_{i,2}, \ldots, v_{i,m})$ is said to satisfy a literal δ if every atom in δ is included in $\{a_1 = v_{i,1}, a_2 = v_{i,2}, \ldots, a_m = v_{i,m}\}$. We define a distribution rule r as $r \equiv \rho(r) \rightarrow (P_1, P_2, \ldots, P_M)$, where its premise $\rho(r)$ is a literal

[1] These assumptions are not necessary but help understanding of a wide range of readers.

and its conclusion is a probabilistic distribution $P_1, P_2, \ldots, P_M$ over the classes $1, 2, \ldots, M$.

A partial decision list T, which may be interpreted as a decision list without the default class label, consists of μ distribution rules $r_1, r_2, \ldots, r_\mu$ i.e. $T \equiv r_1, r_2, \ldots, r_\mu$. For a partial decision list $r_1, r_2, \ldots, r_\mu$, a rule r_j is said to cover an example e iff. (if and only if) e does not satisfy $\rho(r_i)$ $(i = 1, 2, \ldots, j-1)$ but satisfies $\rho(r_j)$. We believe that a partial decision list is adequate as the representation of a hypothesis as it represents a group of rules which are mutually related in a separate-and-conquer manner. The set of examples each of which is satisfied by a distribution rule in a partial decision list T is denoted with $D(T)$.

A null partial decision list B consists of ν distribution rules $b_1, b_2, \ldots, b_\nu$ without conclusions i.e. $B \equiv b_1, b_2, \ldots, b_\nu$. We believe that a null partial decision list is adequate as the representation of an initial hypothesis since it is easier to be obtained from domain experts or textbooks.

We have extended the original MDLP for classification so that T is inferred from D and B. The best hypothesis T_{EMDL} chosen by our extended MDLP is stated as follows.

$$T_{\mathrm{EMDL}} \equiv \arg\min_T \left(-\log P(T) - \log P(D|T) - \log P(B|T)\right)$$

A unique feature of our method is the term $-\log P(B|T)$, which allows us to consider B rigorously. We calculate the code length $L(T)$ in a problem setting where the receiver has D except for the class labels (cf. the sender-receiver problem in the MDLP for classification[30,31]). The sender first sends T, then the class labels of examples in D using T, and B using T.

$$L(T) \equiv -\log P(T) - \log P(D|T) - \log P(B|T) \tag{4}$$

Note that the smaller $L(T)$ is the more interesting T is and thus the negative code length $-L(T)$ can be considered as our interestingness measure. Our measure $L(T)$ returns the degree of the interestingness of a partial decision list T given a data set D and a null partial decision list B as an initial hypothesis. Details of the measure are given in [19].

Our measure has exhibited high "discovery accuracy" i.e. the ratio that the measure discovers the true hypothesis from several artificial data with up to 30 % of noise using incomplete initial hypotheses. Our measure far outperforms the J-measure which is extended to evaluate the interestingness of a partial decision list in most of the cases. Details of the experiments are reported in [19].

4 Concluding Remarks

In this paper, we have introduced J-measure [15] and a measure based on an extended MDLP [19] as examples of compression-based measures for mining interesting rules. The measures exhibit attractive properties including parameter-freeness, robustness against noise, and theoretical bases. The latter can also exploit initial hypothesis and evaluate structured rules i.e. a partial decision tree. We believe that many promising avenues exist for research on compression-based measures for mining interesting rules.

Acknowledgment

A part of this research was supported by the Strategic International Cooperative Program funded by Japan Science and Technology Agency (JST) and the grants-in-aid for scientific research on fundamental research (B) from the Japanese Ministry of Education, Culture, Sports, Science and Technology.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I., et al.: Fast discovery of association rules. In: Fayyad, U.M., et al. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 307–328. AAAI/MIT Press, Menlo Park (1996)
2. Barthélemy, J. P., Legrain, A., Lenca, P., Vaillant, B.: Aggregation of Valued Relations Applied to Association Rule Interestingness Measures. In: Torra, V., Narukawa, Y., Valls, A., Domingo-Ferrer, J. (eds.) MDAI 2006. LNCS, vol. 3885, pp. 203–214. Springer, Heidelberg (2006)
3. Bayardo, R.J., Agrawal, R.: Mining the Most Interesting Rules. In: Proc. Fifth ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, pp. 145–154 (1999)
4. Brin, S., Motwani, R., Silverstein, C.: Beyond Market Baskets: Generalizing Association Rules to Correlations. In: SIGMOD 1997, Proc. ACM SIGMOD Int'l Conf. on Management of Data, pp. 265–276 (1997)
5. Carvalho, D.R., Freitas, A.A., Ebecken, N.F.F.: Evaluating the Correlation Between Objective Rule Interestingness Measures and Real Human Interest. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS, vol. 3721, pp. 453–461. Springer, Heidelberg (2005)
6. Gras, R.: L' Implication Statistique. La Pensée Sauvage (1996) (in French)
7. Jaroszewicz, S., Simovici, D.A.: Interestingness of Frequent Itemsets Using Bayesian Networks as Background Knowledge. In: Proc. Tenth ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, pp. 178–186 (2004)
8. Lenca, P., Meyer, P., Vaillant, B., Lallich, S.: On Selecting Interestingness Measures for Association Rules: User Oriented Description and Multiple Criteria Decision Aid. European Journal of Operational Research 184(2), 610–626 (2008)
9. Liu, B., Hsu, W., Chen, S.: Using General Impressions to Analyze Discovered Classification Rules. In: Proc. Third Int'l Conf. on Knowledge Discovery and Data Mining (KDD), pp. 31–36 (1997)
10. Liu, B., Hsu, W., Mun, L.-F., Lee, H.-Y.: Finding Interesting Patterns Using User Expectations. IEEE Trans. Knowledge and Data Eng. 11(6), 817–832 (1999)
11. Padmanabhan, B., Tuzhilin, A.: A Belief-Driven Method for Discovering Unexpected Patterns. In: Proc. Fourth Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 94–100 (1998)
12. Piatetsky-Shapiro, G.: Discovery, Analysis, and Presentation of Strong Rules. In: Knowledge Discovery in Databases, pp. 229–248. AAAI/MIT Press, Menlo Park (1991)
13. Silberschatz, A., Tuzhilin, A.: On Subjective Measures of Interestingness in Knowledge Discovery. In: Proc. First Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 275–281 (1995)
14. Silberschatz, A., Tuzhilin, A.: What Makes Patterns Interesting in Knowledge Discovery Systems. IEEE Trans. Knowledge and Data Eng. 8(6), 970–974 (1996)

15. Smyth, P., Goodman, R.M.: An Information Theoretic Approach to Rule Induction from Databases. IEEE Trans. Knowledge and Data Engineering 4(4), 301–316 (1992)
16. Suzuki, E.: Autonomous Discovery of Reliable Exception Rules. In: Proc. Third Int'l Conf. on Knowledge Discovery and Data Mining (KDD), pp. 259–262 (1997)
17. Suzuki, E.: Undirected Discovery of Interesting Exception Rules. Int'l Journal of Pattern Recognition and Artificial Intelligence 16(8), 1065–1086 (2002)
18. Suzuki, E.: Pitfalls for Categorizations of Objective Interestingness Measures for Rule Discovery. In: Gras, R., Suzuki, E., Guillet, F., Spagnolo, F. (eds.) Statistical Implicative Analysis: Theory and Applications, pp. 383–395. Springer, Heidelberg (2008)
19. Suzuki, E.: Negative Encoding Length as a Subjective Interestingness Measure for Groups of Rules. In: Proc. Thirteenth Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD (2009) (accepted for publication)
20. Suzuki, E., Shimura, M.: Exceptional Knowledge Discovery in Databases Based on Information Theory. In: Proc. Second Int'l Conf. Knowledge Discovery and Data Mining (KDD), pp. 275–278 (1996)
21. Tan, P.N., Kumar, V., Srivastava, J.: Selecting the Right Interestingness Measure for Association Patterns. In: Proc. Eighth ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD), pp. 32–41 (2002)
22. Vaillant, B., Lallich, S., Lenca, P.: On the Behavior of the Generalizations of the Intensity of Implication: A Data-driven Comparative Study. In: Gras, R., Suzuki, E., Guillet, F., Spagnolo, F. (eds.) Statistical Implicative Analysis: Theory and Applications, pp. 421–447. Springer, Heidelberg (2008)
23. Suzuki, E.: Interestingness Measures - Limits, Desiderata, and Recent Results. In: Proc. Quality Issues, Measures of Interestingness and Evaluation of Data Mining Models, QIMIE (2009) (keynote talk, accepted for publication)
24. Suzuki, E.: Evaluation Scheme for Exception Rule/Group Discovery. In: Intelligent Technologies for Information Analysis, pp. 89–108. Springer, Heidelberg (2004)
25. Keogh, E.J., Lonardi, S., Ratanamahatana, C.A.: Towards Parameter-free Data Mining. In: Proc. Tenth ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD), pp. 206–215 (2004)
26. Feldman, R., Dagan, I.: Knowledge Discovery in Textual Databases (KDT). In: Proc. First International Conference on Knowledge Discovery and Data Mining (KDD), pp. 112–117 (1995)
27. Keogh, E.J., Pazzani, M.J.: Scaling up Dynamic Time Warping to Massive Dataset. In: Żytkow, J.M., Rauch, J. (eds.) PKDD 1999. LNCS (LNAI), vol. 1704, pp. 1–11. Springer, Heidelberg (1999)
28. Blachman, N.M.: The Amount of Information That y Gives About X. IEEE Transactions on Information Theory IT-14(1), 27–31 (1968)
29. Hájek, P., Havel, C.M.: The GUHA Method of Automatic Hypotheses Determination. Computing 1, 293–308 (1966)
30. Quinlan, J.R., Rivest, R.L.: Inferring Decision Trees Using the Minimum Description Length Principle. Information and Computation 80(3), 227–248 (1989)
31. Wallace, C.S., Patrick, J.D.: Coding Decision Trees. Machine Learning 11(1), 7–22 (1993)

Optimization of Feature-Opinion Pairs in Chinese Customer Reviews

Yongwen Huang[1], Zhongshi He[1], and Haiyan Wang[2]

[1] College of Computer Science, Chongqing University, Chongqing 400044, China
[2] Art History Department, Sichuan Fine Arts Institute, Chongqing 400052, China

Abstract. Customer reviews mining can urge manufacturers to improve product quality and guide people a rational consumption. The commonly used mining methods are not satisfactory in precision of the features and opinions extracting. In this paper, we extracted the product features and opinion words in a unified process with semi-supervised learning algorithm, and made an adjustment of the threshold value of confidence to obtain a better mining performance, then adjusted the features sequence with big standard deviation, and maximized the harmonic-mean to raise the precision while ensured the recall. The experiment results show that our techniques are very effective.

Keywords: Customer reviews, Reviews mining, Bootstrapping, Harmonic-mean.

1 Introduction

Customer reviews are more concerned about the using experience compared with the Specifications and the Editor Evaluations, so the mining of customer reviews can help people to buy a suitable product and urge manufacturers to improve their products. It is hard to do the research on Chinese customer reviews mining for the lack of mature reviews corpus and complete evaluation system. In this paper, we first established a mobile phone customer reviews corpus on the basis of the reviews on several Chinese websites.

There have been some researches in product features and opinion words identification. In English customer reviews mining, Soo Min Kim searched for the opinion words basing on the appraise dictionary which defined artificially, and then analyzed the sentence semantic structure with the FrameNet to identify the evaluation targets[1]. Nozomi Kobayashi extracted the names, features and opinion words from the car reviews through the artificial defined pattern <Attribute> of <Subject> is <Value> and the product dictionary, the features dictionary and the opinion dictionary [2]. According to IMDB, Li Zhuang defined the film features' types, features and opinion words artificially, and took the syntax path between them as a template to extract feature–opinion pairs [3]. Hu and Liu first parsed the subjective comments to find the nouns or noun phrases, then extracted frequent features with the association rules, and took the nearest adjective of the product feature as an opinion word [4]. Ana-Maria Popescu utilized statistic-based point-wise mutual information (PMI) to extract product features basing on the association of opinion words with product

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 747–756, 2009.

features, they took the advantage of the syntactic dependencies computed by the MINIPAR parser to identify opinion words[5].In Chinese customer reviews mining, Lun-Wei Ku extracted the opinion words from the sentences with the use of GI (General Inquirer), CNSD (Chinese Network Sentiment Dictionary), NTUSD (National Taiwan University Sentiment Dictionary), which were expanded with the synonymous words and WORDNET[6]. Zheng Yu proposed a statistical method to extract product features from Chinese customer reviews. It is based on distribution of a candidate word in different domains and within the certain domain [7].

Most of the above researches separated the identification of product features and opinion words. In fact, the product features are modified by the opinion words in the context, and there is a close correlation between them, so we identify them in a unified process. Moreover, there are match relations between the product features and the opinion words. Different product features are in general modified by different opinion words, so wrong identification is inevitable in mining. The above studies hadn't handled it. We optimize the sequence by removing some wrong elements to enhance the precision of the identification.

We mainly use Bootstrapping to extract the feature-opinion pairs, then study the change of the precision and the recall with different threshold value of confidence. Finally, we optimize the sequences of opinion words with the maximization of the harmonic-mean to improve the precision while ensure the recall.

2 Bootstrapping

Bootstrapping is a semi-supervised machine learning method [7]. This approach has the advantage of providing only a handful of seeds, the system will automatically extract the text patterns. Then it extracts the candidate seeds in corpus with the patterns, and selects the candidate seeds with high confidence into the seed set. The process of identification is shown in Fig. 1.

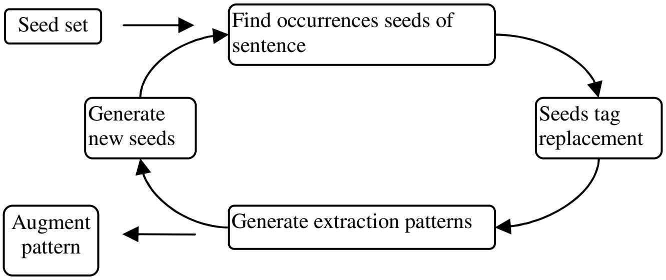

Seed set
Find occurrences seeds of sentence
Generate new seeds
Seeds tag replacement
Augment pattern
Generate extraction patterns

Fig. 1. The flow chart of Bootstrapping

In Fig.1, the seed set is a small amount of labeled data. Bootstrapping finds the sentences with the seeds in the corpus, and generalizes the specific seed with a common tag replacement. Then it generates the candidate text patterns according to the feature set, and adds the candidate pattern with the high confidence to the pattern set. Then it searches the candidate seeds from corpus with the patterns, adds the candidate seeds with the highest confidence to the seed set, and further iterate. In the course of iteration, the selection of the candidate patterns and seeds is the most important. The best way to evaluate a pattern is the number of seeds.

The confidence of a candidate pattern is:

$$conf(p_i) = \frac{S(p_i)}{M} \tag{1}$$

$conf(p_i)$ is the confidence of pattern p_i , $S(p_i)$ is the number of candidate seeds in the seed set searched from corpus by pattern p_i, M is the total number of the searched candidate seeds by patter p_i. Formula (1) mainly takes account of the pattern precision. The pattern with high precision will be able to get good seeds, and will not cause a sharp lower performance after many times of iterations.

The confidence of a candidate seed is:

$$conf\,(F) = 1 - \prod_{i=1}^{n} (1 - conf\,(p_i)) \tag{2}$$

In formula(2), $conf(F)$is the confidence of candidate seed F , n is the text pattern set from which extracted the candidate seed F, and $conf\,(p_i)$ is the confidence of pattern p_i. It shows that the more support patterns a new candidate seed obtains, and the higher confidence the pattern is, thus the more reliable the seed is. The candidate seeds with the confidence higher than the threshold value are added to the seed set for further iteration.

3 Emotion Analysis of Chinese Customer Reviews

The extensive use of the network has caused a large increasing amount of reviews. Our experimental data was downloaded from http://bible.younet.com/, and deleted the contents which are not customer evaluation.

3.1 Analysis of Customer Reviews

From the received reviews, we learn that there are several features in customer reviews. First, the description is dishevelled. It is just about an evaluation on the screen, may soon be about a comment on the software, such as “外形漂亮，滑盖设计的也不错，电池待机时间也还算满意(*the shape is beautiful, the slide is also well-designed, the battery standby time is satisfying*)”. Second, the signification of the feature words is not clear. It may represent a variety of product features, such as "效果(*effect*),性能(*performance*),速度(*speed*)" can not be seen as the features. It needs to extract the former contents as the real evaluated target. For "照相、MP3、视频播放，效果都是一般(*taking pictures, MP3 playing, Video playing, the effect is just so so*)", we should extract the former words"照相、MP3、视频播放(*taking pictures, MP3 playing, video playing*)". Third, the emotion expression is very brief. In terms of "操作非常方便、

电池接触不好、通话声音有点小、性价比高(*the operation is very convenient, the battery can not contact well, the telephone's sound is a little low, the cost-effective is high*)", there are no words before the product features or behind the opinion words. In such comments, few words can be used as a pattern of the features.

Compared with English reviews, the expression is quite different in Chinese reviews. First of all, there may be no articles before the feature words while it is necessary in English. Moreover, there can be no other words between feature words and opinion words while there must be a verb in English. Thus the patterns should be different in Chinese reviews mining. Therefore, we choose words, part of speech, and the modified relationship as the feature set.

3.2 Extraction of Feature-Opinion Pairs

There is corresponding relationship between the evaluated target and the evaluating word. It would be lost if we extract the opinion words and the product features separately. In this paper, we extract them in a unified process like Eugene[8] and Brin[9]. Eugene extracts the relationship between organization and location while Brin extracts the relationship between the author and the book title in a pattern. The pattern structure in this paper as follows:

<pre-modifier of feature and pos, prepositions and pos> tag1 <mid-contents >tag2 <postposition and pos, the post-modifier and pos>

tagl and tag2 respectively refer to product features and opinion words, <mid-contents> stands for the words, pos, the interdependent relationship between the product features and the opinion words.

We firstly find out the sentences with a given feature-opinion pair seed in the reviews corpus, and patternize the obtained sentences with the pattern features set to obtain the candidate pattern. Then find the sentences with the candidate pattern in corpus, and extract the candidate seeds. We calculate the confidence of the candidate pattern with formula (1), and add the pattern with higher confidence to the pattern set. Then the candidate seeds' confidence is calculated by formula (2), and the seed with higher confidence is added to the seed set. This process is iterated to mine new feature-opinion pairs.

3.3 Optimization of Feature-Opinion Pairs

The precision of the feature-opinion pairs which extracted by Bootstrapping is not high. There are some wrong recognition in the result, such as wrong product features, wrong opinion words and wrong relations between them. Compared with the correct recognition, the number of the wrong identification is small, so we deal with the sequences of product features and opinion words by optimizing the sequence to remove some wrong elements, so as to enhance the precision of the identification.

The sequences of opinion words can be divided into two groups. One group are those change in the sequence is not obvious, such as "好(*good*), 一般(*just so so*), 不错(*not* *bad*)" can be used to evaluate most product features, their sequences need not to be handled. The others' change in the sequence is very obvious, such as"清晰(*clear*), 耐用(*durable*), 性价比(*cost-effective*)" can be only respectively used to evaluate the screen, the battery, and the price. If the identification is wrong, the value would be low relatively and the sequence elements value would also have a great change. We use the standard deviation of the sequence to determine the modification of opinion words, and optimize the sequence with a big standard deviation to improve the mining precision.

First, calculate the standard deviation S for each sequence of opinion words:

$$S = \sqrt{\tfrac{1}{n}[(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + ... + (x_n - \bar{x})^2]} \tag{3}$$

In formula (3), n is the length of the sequence, x_1,x_2,...x_n are the value of the sequence, and $\bar{x}$ is the average value of the sequence.

The criterion of optimization is maximizing harmonic-mean of the results, making the precision as high as possible while ensuring the recall. The precision and the recall are defined by formula (4):

$$\mathrm{Pr}\,ecision(A) = \frac{N(c)}{N(a)} \qquad \mathrm{Re}\,call(R) = \frac{N(c)}{N(r)} \tag{4}$$

Where $N(c)$ is the number of the correct feature-opinion pairs, $N(a)$ is the number of all the extracted feature-opinion pairs, $N(r)$ is the actual number of the feature-opinion pairs in the testing set.

The harmonic-mean $F1$ is:

$$F1 = \frac{2}{\frac{1}{P} + \frac{1}{R}} = \frac{2PR}{P + R} \tag{5}$$

The harmonic-mean is vulnerably impacted by the extreme value, especially the minimum value, so maximize the value of $F1$ can ensure both the precision and the recall.

4 Experiments and Analysis

Chinese reviews on website look like a messy compared with English reviews. For the comments on the same product, Nokia n95, on Amazon website both in English and in Chinese, it is almost the same in quantity, while it is quite disparate in quality. In Chinese reviews, there are rare experiences with the product purchasing, using and after-sales service. While English reviews are mainly about customers' subjective evaluation on the product performance, function, accessories and after-sales service etc, so they can be used as a corpus directly. It needs us to make more effort on the mining of Chinese reviews.

Our experimental data were from http://bible.younet.com. It mainly introduces phone information, and its reviews are selected by specialist. First of all, we extracted the latest 5000 reviews about 500 pages from1700 pages, and then classified them by phone model. Bootstrapping needs the reviews as much as possible, so we took reviews about 20 models more than 100K, 18 of them are as a training corpus, 2 are as a test corpus. Firstly, we handled the corpus for segments and part-of-speech tagging with ICTCLAS[1](Chinese lexical analysis system), and then marked out the modifying relations with the ICTPROP(Chinese probability Parser), and labeled the feature-opinion pairs in the test set artificially.

4.1 Selection of Confidence Threshold Value

When we providing seeds and threshold value of confidence, Bootstrapping method can automatically extracte the patterns and the seeds from the corpus. Bad seeds and patterns can result in the iteration performance reducing sharply, the threshold value has great influence on the last result. In this paper, we make a appropriate confidence threshold value in Chinese reviews corpus.

We artificially choose 40 feature-opinion pairs about screen, sound, battery as the seeds from phone reviews, such as“性价比-高(*cost effective–high*), 声音-大(*voice–loud*),速度-慢(*speed–slow*), 识别率-高(*recognition rate–high*), 屏幕-清晰(*screen–clear*),音质-好(*sound quality–good*), 电池-耐用(*battery–durable*),信号-好(*signal–fine*),键盘-宽大(*keyboard- roomy*)”. Then we use the word, part of speech and the modification relationship as the features in a pattern, designate the threshold value for 10 times iteration in training corpus. After the comparison of the feature-opinion pairs which mining from test set with those artificially labeled, we get the precision and the recall under the given threshold value. Then designate a different value and mine again. As shown in figure 2, the precision and recall change with the threshold value.

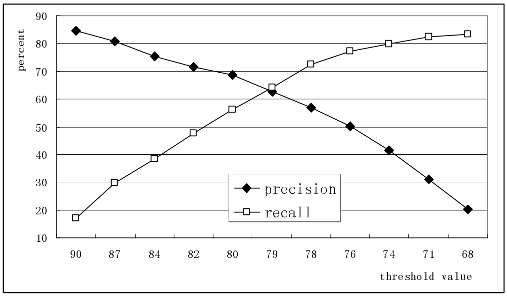


Fig. 2. The change of precision and recall with the change of threshold value

It shows that when the threshold value is higher, there is a higher precision but a lower recall. Due to the high threshold value, only a few candidate patterns can be added into the pattern set, and also only a few candidate seeds can join the seed set. Many correct seeds are excluded inevitably, so the recall is very low. As the threshold value goes down, more candidate patterns and seeds can join the pattern set and the seed set. These added patterns and seeds may not correct thus caused the precision

[1] http://www.nlp.org.cn/

dropped. At the same time, more correct feature-opinion pairs may be extracted from the reviews, so the recall will increase. Although precision is very important, but what we need in the learning algorithm is to mine new feature-opinion pairs, so we set the precision about 50 percent, recall about 70 percent, threshold value with 77.

4.2 Extraction of Feature-Opinion Pairs

Based on the extraction of the training corpus with Bootstrapping, the patterns is obtained which were able to extract the seed and a number of new feature-opinion pairs, as shown in Table 1.

Table 1. The patterns and new feature-opinion pairs

Some patterns	Feature-opinion pairs	
feature 还是/c 非常/d opinion 的/u	成像质量	差
feature 还是/c 相当/v opinion 的/u	性价比	値
feature 感觉/n 还/d opinion	音效	出色
feature 功能/n 也/d 很/d opinion	屏幕	粗糙
感觉/n feature 还是/c 有点/d opinion	屏幕色彩	棒
不过/c feature只能/v算/v opinion.	视频播放	流畅
feature 确实/a很/d opinion	视觉效果	震撼
最/d opinion 的/u是/v feature	强	铃声

Table 1 shows that the form of the pattern is basically very simple and many patterns have only features and opinion words. The users usually express their opinions in a very short way, so there is no other content can be served as a judge on the product features and opinion words. It increases the difficulty in Chinese customer reviews mining.

4.3 Optimization of Sequences

The identification result of Bootstrapping is not satisfactory, for the precision is no more than 60 percent, and there are some errors in the low-frequency feature-opinion pairs. The sequence optimization can adjust the wrong data to improve the precision of the experimental result.

We remove the feature words with low-frequency emergence first. In a test corpus about 300,000 words and 130,000 phrases, if a product feature appeared few, it indicates that the product feature is not concerned by the customer, so it can be ignored in extracting. In this paper, we deleted all the product features which appeared less than 5 times.

Calculate the standard deviation of the opinion word sequence with formula (3), only non-zero sequence of elements considered. Then optimize the sequences which standard deviation is more than 5. The optimization processes are: First calculate the sum of all the opinion words sequence, then delete the elements whose frequency is less than the number multiplied by the percentage of the total number, gradually increase the deleting proportion and calculate the harmonic-mean F1 at any time. The results are shown in Table 2.

Table 2. The changes of evaluating indicator with the remove of the low-frequency elements

deleting proportion	0%	delete lows	3%	5%	6%	7%	8%	9%
precision(%)	52.37	57.68	61.29	64.63	67.5	70.12	72.17	73.96
recall(%)	74.36	74.18	73.64	73.13	72.51	71.82	70.96	69.97
F1(%)	61.46	64.90	66.90	68.62	69.92	70.96	71.56	71.91
deleting proportion	**10%**	11%	12%	13%	14%	15%	16%	17%
precision(%)	**75.62**	76.54	77.13	77.64	78.09	78.45	78.69	78.85
recall(%)	**68.91**	67.47	65.49	63.04	60.20	56.29	51.50	44.56
F1(%)	**72.11**	71.72	70.83	69.58	67.99	65.55	62.26	56.94

“delete lows” means deleting the feature which frequency is lower than 5

Table 2 shows that after the deletion of the low-frequency elements, the precision and recall of the obtained feature-opinion pairs have changed a lot. At the beginning of the deletion, the precision rises quickly, the recall declines slowly, and the harmonic-mean F1 gradually increases. With the increasing of the deleting proportion, the precision increases slower, the recall drops sharply, and the harmonic-mean F1 attains the maximum when the deleting proportion is at 12 percent, then it reverses down as the deleting proportion increases. The changes are shown in Fig. 3.

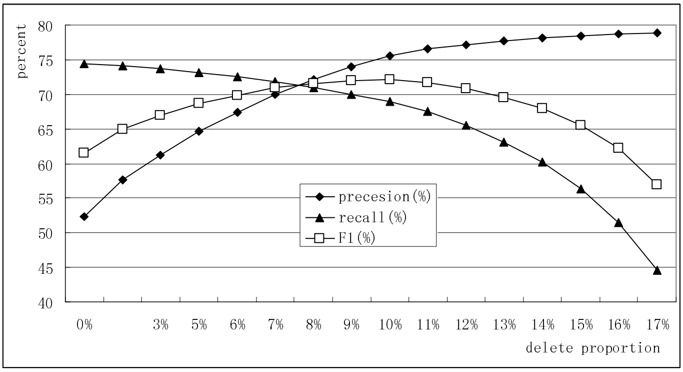


Fig. 3. The changes of evaluating indicator with deleting proportion

In Table 2 and Fig.3, the removal of low-frequency features leads the precision to increase about 5 percent, while the recall drop down about only 0.2 percent. It means that these low-frequency features can be deleted. With the removal of the opinion words sequences of low-frequency, the recall decreases gradually, but the precision increases even faster, and the harmonic-mean *F*1 is also increasing. When the deleting proportion is at 10 percent, *F*1 is the maximum, the precision is near 75.6 percent and the recall is near 68.9 percent, while the precision raises 14 percent and the recall gets down only 6 percent. So it is desirable to remove low-frequency sequence to improve the extraction results of Bootstrapping.

4.4 Experiments Comparison

In this paper, we implemented the association rule mining algorithm of Hu and Liu [4,10], then we used the features pruning to remove the non-adjacent and contained product features, took the nearest adjective of the product feature as its opinion word to form a feature-opinion pair. The experiment results are shown in table 4.

Table 3. Comparison results with Hu and Liu's work

experiment	precision(%)	recall(%)	*F*1(%)
Our work	75.62	68.91	72.11
Hu and Liu's	63.43	52.65	57.54

Table 2 and table 3 tell us, before the sequence optimization, Hu and Liu's association rules algorithm is better than Bootstrapping in mining. It can extract the feature words in describing sentences, but these features may be ignored if extracted by Bootstrapping. Meanwhile, Hu and Liu's method does not have the advantage in the recognition of opinion words and the extraction of features-opinion pairs in Chinese customer reviews. First of all, adjectives in Chinese can be not only an evaluation but also a modifier. Hu and Liu' method only takes the nearest adjective of the feature as the evaluation, it is inevitable to make a wrong feature-opinion pair extraction. Moreover, the use of punctuation in customer reviews is not standard, there may be a comma and quotation marks between the feature words and the opinion words, such as"2.2英寸的Q屏，很细腻(*2.2-inch Q screen, very exquisite*)". So the corresponding information of the feature-opinion pair would miss if we take a comma as a sentence separator. Bootstrapping does not take punctuation as a sentence separator in mining, thus feature-opinion pairs can be accessed remotely. Furthermore, for some opinion words are not adjective, such as "小/a 钢壳/n 比较/d 有/v 手感/n", Hu and Liu's method can not get the opinion. In a word, the methods of English reviews mining are not effective in the mining of Chinese customer reviews.

In this paper, the sequence optimization significantly improved the precision of the reviews mining. The results show that our method is 12 percent higher than Hu and Liu's method.

5 Conclusion

The difference for reviews between Chinese and English is analyzed in this paper. Words, part of speech and modified relations is chosen to form the feature set of the patterns which can be extracted feature-opinion pairs. Meanwhile, the threshold value of the confidence with Bootstrapping is discussed, the product features and opinion words is extracted from the customer reviews in a unified process. In the end, we removed the wrong identifications of product features, opinion words and feature-opinion pairs with the removal of the low-frequency feature sequences, and maximized the harmonic-mean to improve the precision while ensured the recall was not too low. The results showed that the treatment is effective.

Obtaining of the feature-opinion pairs is only the first step in reviews mining. We carry out some further research work about giving an appraise evaluation judge on the obtained opinion words, making a statistics of the customer evaluations, graphically displaying the emotional expression of each product feature, and the application of customer reviews mining to actual practice.

Acknowledgement

This work is supported by the Natural Science Foundation Project of Chongqing CSTC 2007BB2134, and the Chongqing Higher Education Reform Subject Research Project 0635207.

References

1. Kim, S.M., Hovy, E.: Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text. In: Proceedings of the ACL/COLING Workshop on Sentiment and Subjectivity in Text, Sydney, Australia, pp. 1–8 (2006)
2. Kobayashi, N., Inui, K., Matsumoto, Y., Tateishi, K., Fukushima, T.: Collecting Evaluative Expressions for Opinion Extraction. In: Su, K.-Y., Tsujii, J., Lee, J.-H., Kwong, O.Y. (eds.) IJCNLP 2004. LNCS, vol. 3248, pp. 584–589. Springer, Heidelberg (2005)
3. Zhuang, L., Jing, F., XiaoYan, Z.: Movie Review Mining and Summarization. In: Proceedings of the 15th ACM international conference on Information and Knowledge Management, pp. 43–50 (2006)
4. Minqing, H., Bing, L.: Mining Opinion Features in Customer Reviews. In: Proceedings of Nineteenth National Conference on Artificial Intelligence (AAAI 2004), San Jose, USA, pp. 755–760 (2004)
5. Popescu, A.M., Etzioni, O.: Extracting Product Features and Opinions from Reviews. In: HLT/ EMNLP, pp. 339–346 (2005)
6. Lun-Wei, K., Hsin-His, C.: Mining Opinions from the Web: Beyond Relevance Retrieval. Journal of the American Society for Information Science and Technology 58(12), 1838–1850 (2007)
7. Yu, Z., Liang, Y., Gengfeng, W., Xin, L.: Extracting Product Features from Chinese Customer Reviews. Intelligent System and Knowledge Engineering 1, 285–290 (2008)
8. Agichtein, E., Gravano, L.: Snowball: Extracting Relations from Large Plain-Text Collections. In: ACM International Conference on Digital Libraries, pp. 85–94. ACM Press, New York (2000)
9. Brin, S.: Extracting Patterns and Relations from the World Wide Web. In: International Workshop on the Web and Databases Spain, pp. 172–183 (1999)
10. Liu, B., Hsu, W., Ma, Y.: Integrating Classification and Association Rule Mining. In: KDD 1998, pp. 80–86 (1998)

iTM: An Efficient Algorithm for Frequent Pattern Mining in the Incremental Database without Rescanning

Bi-Ru Dai and Pai-Yu Lin

Department of Computer Science and Information Engineering,
National Taiwan University of Science and Technology, Taipei, Taiwan, ROC
brdai@csie.ntust.edu.tw, m9615082@mail.ntust.edu.tw

Abstract. Frequent pattern mining plays an important role in the data mining community since it is usually a fundamental step in various mining tasks. However, maintenance of frequent patterns is very expensive in the incremental database. In addition, the status of a pattern changes with time. In other words, a frequent pattern is possible to become infrequent, and vice versa. In order to exactly find all frequent patterns, most algorithms have to scan the original database completely whenever an update occurs. In this paper, we propose a new algorithm iTM, stands for incremental Transaction Mapping algorithm for incremental frequent pattern mining without rescanning the whole database. It transfers the transaction dataset to the vertical representation such that the incremental dataset can be integrated to the original database easily. As demonstrated in our experiments, the proposed method is very efficient and suitable for mining frequent patterns in the incremental database.

Keywords: Frequent Patterns, Incremental Database, Incremental Mining.

1 Introduction

The frequent pattern plays a key role in many data mining tasks, such as association rules [1,4,10], sequential patterns [11], classification [2,3], and clustering [4]. The frequency of a pattern is computed by counting its occurrence in the transaction database. If the frequency exceeds a user defined threshold, the pattern is considered to be important in the transaction database. However, the cost of finding all frequent patterns is expensive, because patterns are generated from combinations of items, and the number of combinations is enormous. Therefore, mining frequent patterns efficiently becomes a difficult challenge and attracts lots of research attention.

While many efficient algorithms are proposed in recent years, the amount of data also has a huge growth. The size of database evolves from megabytes to terabytes in recent two decades. Furthermore, hundreds of megabytes of new datasets are accumulated to the database every day. Even though many efficient algorithms have been proposed, mining frequent patterns iteratively without past frequent patterns is still very expensive. Therefore, developers change their focus to incremental frequent patterns mining [6,7,8,9,11]. Cheung et al. develop an algorithm, named FUP [6] which is the first incremental mining algorithm. It reuses information from old frequent patterns to save the computation cost. Mining frequent patterns in the

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 757–766, 2009.

incremental database is more complicated than the mining of the static transaction database. Some frequent patterns are possible to become useless after more data records are collected. Infrequent patterns encounter the similar situation. These phenomena bring about new challenges, such as counting the frequency of a pattern in the updated database.

In order to find frequent patterns from the incremental transaction database, most incremental approaches have to rescan the original database for counting frequencies of new frequent patterns, and then it becomes the bottleneck in this type of approaches. Hence, earlier incremental approaches focus on reducing the number of rescanning the original database when an update occurs. Some developers also try to solve the trade-off in accuracy and efficiency [25]. The approach allows incorrect result in frequent patterns, but guarantees the error bound. It effective increases the efficiency for mining frequent patterns in some situation, if user does not request the accuracy.

One of the algorithms had been proposed for incremental mining without rescanning the original database is the CanTree algorithm [12]. CanTree can be easily maintained without adjusting, merging, and splitting tree nodes when the transaction database is updated. However, it incurs another issue that balances performance between maintenance of supports of patterns and efficiently mining frequent patterns. Unfortunately, they are exclusive in most situations or methods.

In this paper, we propose a refined algorithm of incremental frequent pattern mining base on TM algorithm [5]. It is called *incremental Transaction Mapping*, abbreviated as iTM. This method maintains frequent patterns in the incremental database without rescanning the original database completely. We also focus on achieving a better compression ratio on the representation of the whole database. In addition, iTM has other benefit that scanning the database only once for constructing its compressed data structure while an update occurs and the data structure can be reused in interactive mining. This is an important feature of our method: building once and mining multiple times. Experiment presents that iTM has the ability of mining frequent patterns in the incremental transaction database efficiently. In addition, it demonstrates that the representation of the transaction database can effectively assist in frequent patterns mining.

The rest of this paper is organized as follows. Section 2 describes some related works about incremental mining. The problem definition will be given in section 3. Then we discuss our algorithm iTM in section 4. Section 5 presents performance evaluation of our method. Conclusion will be given in Section 6.

2 Related Work

In the past, many developers had reported the issue of incremental data mining. The focus of this issue is on maintaining frequent patterns when the database is updated. The first algorithm is FUP which is proposed by Cheung, D.W. et al. [6]. The idea of FUP is to store the supports of all frequent patterns in the previous mining process. Then, this information is used to check status of frequent patterns when an update occurs. The FUP iteratively discovers new frequent patterns in Apriori-like manner from the updated database. Even though, FUP stores the supports of frequent patterns, it must scan the original database several times for obtaining supports of infrequent patterns. Another method is proposed by Thomas, S. et al. [13]. It is based on the concept of Negative Border. The algorithm maintains the negative borders of the patterns in the main memory. It efficiently reduces the times of scanning the whole database to ascertain frequent patterns in the updated database. The SWF is proposed by Lee, C.H. et al. [14]. It adopts the technique of sliding-window. It saves the memory space and efficiently increases speed of execution.

Another type of algorithms is also proposed in recent years. FELINE [15], AFPIM [16] and CanTree [12] use different concepts for incremental data mining. They store all information of patterns in the data structure, such as FP-Tree and CanTree. FELINE algorithm is proposed by Cheung, W. et al. [15]. They design the CAST tree mainly for interactive mining. The CAST tree extends the idea of the FP-Tree to improve the ratio of compression. AFPIM method is proposed by Koh, J.L. et al. [16]. They adopt the concept of pre-minsup and employ the original notion of FP-tree. However, selecting a suitable pre-minsup is a difficult problem. The CanTree is proposed by Leung, C.K.S. et al [12]. It makes the maintenance of data structure more easily when update occurs. FELINE and AFPIM algorithm have to adjust the tree structure, such as merge and split, while CanTree only has the insert step when update occurs. Therefore CanTree has less execution time for maintenance of data structure.

TM algorithm [5] is a technique to compress *TID*s (transaction ids). The representation of the transaction database has two types, horizontal and vertical. TM algorithm is the second representation, and uses continue intervals to reach good compression ratio. TM algorithm has good performance on the balance of space and time cost. Unfortunately, this method only supports mining frequent patterns in static transaction databases. Therefore, we provide a method based on TM algorithm in this work. Our objective is to keep advantages of TM algorithm and further extend to support incremental mining.

3 Problem Description

In this section, we introduce the problem of mining frequent patterns in the incremental database. This problem is similar to mining frequent patterns in static databases. However, the patterns are possible to change their status. Below we give formal definitions of mining frequent patterns and the updated of frequent patterns.

3.1 Mining Frequent Patterns

Let $I = \{i_1, i_2, ..., i_m\}$ be a set of items, and *DB* be a transaction database. Each transaction *T* is a set of items such that $T \subseteq I$. The *TID* is the transaction identifier for each transaction. Assume *X* is a pattern and a transaction *T* contains the pattern *X* if and only if $X \subseteq T$. The *support* of the pattern *X* is defined as the percentage of transactions containing the pattern *X*. It is denoted as *X.support*. Given the user-defined minimum support (*minsup*) threshold *s* for *DB*, the frequent pattern mining will find all patterns whose supports do not less than *s*. The rest of patterns are called infrequent patterns.

Table 1. Notation table

Notation	Definition
$\|DB\|$	The number of transactions in the original transaction database
$\|db\|$	The number of transactions in the incremental transaction dataset
$\|DB'\|$	The number of transactions in the updated transaction database
L_{DB}	The set of frequent patterns in *DB*
L_{db}	The set of frequent patterns in *db*
$L_{DB'}$	The set of frequent patterns in *DB'*
s	Minimum support threshold

3.2 Update of Frequent Patterns

Let *db* be the incremental dataset, and *DB'* be the updated database that includes the original database *DB* and the incremental dataset *db*. Let L_{DB} be the set of frequent patterns in the original database *DB*. By definition, the support of each frequent pattern has to be greater than or equal to *s* in *DB*. After an update occurs, the *db* is added to the original database *DB*. With the same *s*, the pattern *X* is frequent in the updated database *DB'* if and only if the number of transactions *T* containing the pattern *X* is not less than $s*|DB'|$. Notations used in this paper are briefly summarized in Table 1.

Table 2. Four scenarios associated with a pattern in *DB'*

db \ *DB*	Frequent	Infrequent
Frequent	Frequent	Frequent or Infrequent
Infrequent	Frequent or Infrequent	Infrequent

When the transaction database is updated, all patterns must be checked for their status. There are four cases in incremental frequent patterns mining. If a pattern is frequent in both *DB* and *db*, it will be frequent in *DB'* absolutely. In another similar case, if a pattern is infrequent in both *DB* and *db*, it will be infrequent in *DB'* intuitively. The status of patterns in these two cases can be determined immediately. Unfortunately, the other two cases are ambiguous. If a pattern is frequent in *DB*, but infrequent in *db*, it is possible to be frequent or infrequent in *DB'*. If a pattern is infrequent in *DB*, but frequent in *db*, the status of this pattern in *DB'* is not obvious either. In these two cases, we need the exact supports of patterns in *DB* and *db* for mining frequent patterns in the updated database *DB'*. Table 2 shows all scenarios associated with a pattern in *DB'*. Earlier algorithms usually remember the supports of patterns in *DB* if the pattern is frequent. Therefore, if a pattern is infrequent in *DB*, its information is lost such that we can not determine the status of the pattern in the updated database *DB'*. This is the main issue for incremental frequent patterns mining. *How much information should be kept for incremental mining?* However, keeping enough information is a difficult work. In this paper, our method will remember all supports of items in the compressed structure such that the space cost is not expensive. We achieve better compression ratio. Furthermore, the procedure of mining frequent patterns also has a good performance.

Table 3. Sample transaction database

TID	Items
1	1,2,3
2	2,3
3	1
4	1,3
5	1,2,3
6	2,4
7	1
8	2,4
9	1,4
10	1,3
11	3,4,2
12	1,3,4,2

Table 4. (a) Transaction mapping table for the original database. (b) Transaction mapping table for the increment dataset. (c) Transaction mapping table for the updated database.

(a)

Item	Interval List
1	[1,5]
2	[1,2],[6,8]
3	[1,3],[6,6]
4	[7,8]

(b)

Item	Interval List
1	[1,3]
2	[1,1],[4,4]
3	[1,2],[4,4]
4	[1,1],[3,3],[4,4]

(c)

Item	Interval List
1	[1,8]
2	[1,3],[9,12]
3	[1,5],[9,10]
4	[1,1],[6,6],[9,9],[11,12]

4 iTM Algorithm (Incremental Transaction Mapping)

The key idea behind of previous incremental mining algorithms is to reduce the number of time to rescan the original database. In the present study, we develop a new algorithm, iTM. It is an extension of TM algorithm. iTM represents the incremental transaction database in a compressed vertical format. iTM not only easily maintains the compressed structure, but also mines frequent patterns efficiently. In addition, iTM supports the goal: building once and mining multiple times.

The iTM algorithm uses the vertical representation for the transaction database. We compress *TID*s (transaction ids) for each pattern to continuous intervals. At the beginning, the step is similar as TM algorithm [5]. The iTM generates each transaction mapping table by the transaction tree. For example, Table 3 gives a sample transaction database. *TID* 1 to *TID* 8 represent the original database, and *TID* 9 to *TID* 12 are considered as the incremental dataset. The transaction mapping table is shown in Table 4(a) and Table 4(b) for the original database and the incremental dataset, respectively. The form of the interval is *[s, e]*, where *s* is the relabeled start id and *e* is the relabeled end id. The value of *e-s+1* equals to the local frequency of an item. In this process, we keep all interval lists for each item, including infrequent items. Although the continuous interval has a better compression ratio, because each interval only records start id and end id. The interval contains more information in the same space, if the support of an item is larger. It is worthy to keep all interval lists of items for update of frequent pattern step.

After the first procedure, we have two transaction mapping tables for the original dataset and the incremental dataset, respectively. Then we use these two transaction mapping tables to creating the new transaction mapping table for the updated database. It achieves effective utilization of the old information. For each item, our merge procedure is summarized as follows:

Step1: Sort the item by supports in the updated database.
Step2: Find intervals of items from old information (transaction mapping tables for the original database and the incremental dataset.)
Step3: Find the intersectional interval from other items whose supports are greater than the present item.
Step4: Calculate the support of the pattern and reassign a new interval to item.

In step 1, we sort these items by their supports. This step will get the organized interval lists. Because of assigning intervals from the item with the largest support count, continuous intervals will be collected. Then, we find the interval for each item from old transaction mapping tables (the original database and the incremental dataset.) For step 3, we will find intersectional intervals for each item. Assume that interval A is $[s_1,e_1]$ and interval B is $[s_2,e_2]$, the intersectional interval of A and B can be summarized in four scenarios as follows:

1. If s_1 is greater than or equal to s_2, and e_1 is greater than or equal to e_2, the intersectional interval is $[s_2, e_1]$.
2. If s_2 is greater than or equal to s_1, and e_2 is greater than or equal to e_1, the intersectional interval is $[s_1, e_2]$.
3. If s_1 is greater than or equal to s_2, and e_2 is greater than or equal to e_1, the intersectional interval is $[s_2, e_2]$.
4. If s_2 is greater than or equal to s_1, and e_1 is greater than or equal to e_2, the intersectional interval is $[s_1, e_1]$.

In here, we only need to find the intersectional interval from the other items whose support is greater than the present item. The other intersectional intervals will be find by the other items whose support is less than the present item. Therefore, all intersectional intervals will not be missed.

In step 4, we calculate supports for patterns and reassign a new interval that covers the most number of items. This step will repeat iteratively until all patterns are covered by new intervals. After this step, the new transaction mapping table of the updated database is completed. To follow, we will mine frequent patterns from the new transaction mapping table. The representation of the transaction database is organized. We start from the item with smallest support to the item with largest support and calculate supports of patterns by intersecting these interval lists. Additionally, the transaction mapping table can be mined multiple times until another update occurs. It achieves the goal of building once and mining multiple times. Below is an example of our procedure:

Table 5. Information of each item during the merging process

Item	Type	Interval	Intersectional Interval List
1	Original	[1,5]	Null
	Incremental	[1,3]	Null
3	Original	[1,3]	1:[1,3]
		[5,6]	Null
	Incremental	[1,2]	1:[1,2]
		[4,4]	Null
2	Original	[1,2]	3:[1,2],1:[1,2]
		[5,8]	3:[5,6]
	Incremental	[1,1]	3:[1,1],1:[1,1]
		[4,4]	3:[4,4]
4	Original	[7,8]	2:[7,8]
	Incremental	[1,1]	2:[1,1],3:[1,1],1:[1,1]
		[3,3]	1:[3,3]
		[4,4]	2:[4,4],3:[4,4]

Example: Table 5 records interval lists of items and their intersectional interval lists with other items from Table 4(a) and Table 4(b). In the first step, we obtain interval lists of items from old transaction mapping tables of the original database and the incremental dataset. For example, the interval list of item 1 contains [1,5] and [1,3], that of item 3 contains [1,3], [5,6], [1,2] and [4,4]. The rest can be deduced by analogy. The goal of next step is to find all intersectional intervals within the present item. Assume that the present item is item 2, and we search the intersectional interval for interval [1,2]. In here, we need to find the intersectional interval from item 1 and item 3, because support of item 1 and item 3 are greater than or equal to item 2. We will obtain interval [1,2] from item 1 and item 3. Then, we reassign a new interval for each item. At the beginning, the frequency of each pattern is calculated. For instance, we focus on item 4. We get intersection intervals 2:[1,1], 3:[1,1], 1[1,1] in interval [1,1] before. The frequency of pattern {1,2,3,4} is 1 from our calculation. We also have to calculate the frequency of pattern {1,2,3,4} in other intervals of item 4 and accumulate them. This pattern is the longest one containing item 4. Therefore, the new interval is intersection with item 1, 2 and 3. By previous steps, we get interval lists of item 1, 2 and 3. They are recorded in Table 4(c). Hence, the new interval starts from 1, and the range of this interval equals the frequency of the pattern. So the new interval is [1,1]. We can check from this equation: the value of *e-s+1* is the frequency of pattern. 1-1+1=1 is equal to the frequency of pattern {1,2,3,4}.

5 Experiment and Performance Evaluation

We conducted extensive experiments to evaluate the performance of the proposed iTM algorithm. We choose a real dataset, chess dataset [17], and simulate it as in the incremental environment. The chess dataset is divided into two parts to simulate the original database and the incremental dataset. The number of transactions in the original database is 1.5K. Remaining transactions are regarded as the incremental dataset. The experimental platform used is a personal computer with a 3G-Hz CPU and 2GB of main memory. The operating system is Windows XP and iTM algorithm was coded by Java version 1.6. The CanTree algorithm is also implemented as the competitor of our algorithm. We separate our experiments into two parts. First, we evaluate the execution time with various sizes of the incremental dataset. The size of the incremental dataset is increased from 0.15K to 1.5K. The *minsup* is set to 50% in this experiment. Second, the execution time is evaluated when varying the minimum support threshold. The *minsup* is varied from 90% to 10%, and the size of the increment dataset is 1.5K. These two experiments evaluate the execution time for two effective factors, the size of the incremental dataset and the minimum support threshold.

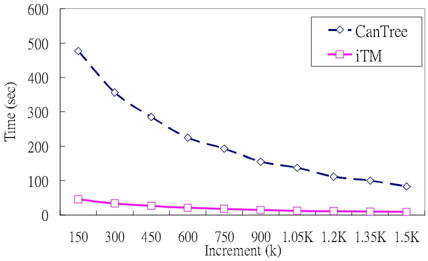


Fig. 1. Comparing execution time when varying the size of incremental dataset (chess data, |*DB*|=1.5K, minsup=50%)

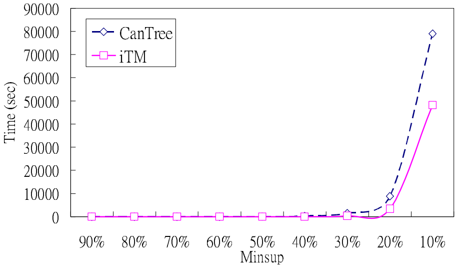


Fig. 2. Comparing execution time when varying the minimum support threshold (chess data, |*DB*|=1.5K, |*db*|=1.5K)

Fig 1 shows the experimental result of constructing the data structure. The lines scratch the relationship between the execution time and the size of the incremental dataset. Table 6 shows detailed information of mining result on chess dataset and execution time in building the data structure. The first column records the number of transactions in the incremental dataset. The second column records the number of frequent patterns in the updated database. The third and forth column records the time for building the data structure in iTM and CanTree respectively. Table 6 demonstrates that the time for building increases along with the size of the incremental dataset. However, the execution time decreases along with the size of the incremental dataset in Fig 1. In the Table 6, the number of frequent pattern is decreased from 0.15K to 1.5K in the updated database. And mining frequent patterns is the most cost in iTM and CanTree. Therefore, the tendency of execution time is reduced, and the ratio of growth is linear in time for building data structure. Note that iTM significantly reduces the execution time of the mining step. As shown in Fig 1, iTM outperforms CanTree for about 10 times.

Table 6. Summary of the relation between the size of db, execution time for building data structure, and number of frequent patterns generated

$\|db\|$	$\|L_{DB'}\|$	Time for iTM (msec)	Time for CanTree (msec)
0.15K	13057119	469	31
0.30K	8633751	515	78
0.45K	6443079	562	141
0.60K	4635455	594	203
0.75K	3609767	687	344
0.90K	2701431	797	625
1.05K	2257943	891	688
1.20K	1831233	1031	828
1.35K	1487858	1141	1156
1.50K	1272932	1250	1469

Fig 2 shows the result of mining frequent patterns while varying the minimum support threshold. The lines scratch the relationship between the execution time and the percentage of *minsup*. In this experimental result, iTM achieves better performance than CanTree. Because items are sorted by the support in iTM, the speed of mining frequent patterns can be reduced. On the contrary, CanTree does not maintain items in a sorted structure. In the mining step, CanTree has to traverse all sub-trees to extract a specific item. As shown in Fig 2, iTM is more efficient than CanTree.

6 Conclusion

In this paper, we propose an algorithm iTM that is based on TM algorithm. We exploit advantage from TM algorithm and design a framework for incremental frequent patterns mining. The contribution of iTM is providing a solution for frequent patterns mining in the incremental transaction database without rescanning the whole database. It reduces the cost of time and space simultaneously. The experimental result demonstrates that iTM is efficient and effective for mining frequent patterns in the incremental database.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules Between Sets of Items in Large Databases. In: 1993 ACM SIGMOD Conference, ACM SIGMOD, Washington, DC, pp. 207–216 (1993)
2. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
3. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth International Group (1984)
4. Lloyd, S.P.: Least Squares Quantization in PCM. IEEE Transactions on Information Theory it-28(2), 129–137 (1982)
5. Song, M., Rajasekaran, S.: A Transaction Mapping Algorithm for Frequent Itemsets Mining. IEEE Transactions on Knowledge and Data Engineering 18(4), 472–481 (2006)
6. Cheung, D.W., Han, J., Ng, V.T., Wong, C.Y.: Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique. In: 12th International Conference on Data Engineering, ICDE, New Orleans, pp. 106–114 (1996)
7. Cheung, D.W., Lee, S.D., Kao, B.: A General Incremental Technique for Maintaining Discovered Association Rules. In: Fifth International Conference on Database Systems for Advanced Application, DASFAA, Melbourne, pp. 185–194 (1997)
8. Chang, C.C., Li, Y.C., Lee, J.S.: An Efficient Algorithm for Incremental Mining of Association Rules. In: 15th International Workshop on Research Issues in Data Engineering: Stream Data Mining and Applications, RIDE-SDMA 2005, pp. 3–10 (2005)
9. Zhang, S., Zhang, J., Zhang, C.: EDUA: An Efficient Algorithm for Dynamic Database Mining. Information Sciences 177, 2756–2767 (2007)
10. Liu, B., Hsu, W., Ma, Y.: Mining Association Rules With Multiple Minimum Supports. In: Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM SIGKDD, San Diego, pp. 337–341 (1999)
11. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C.: PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-projected Pattern Growth. In: 17th International Conference on Data Engineering, ICDE 2001, pp. 215–224 (2001)
12. Leung, C.K.S., Khan, Q.I., Hoque, T.: CanTree: A Tree Structure for Efficient Incremental Mining of Frequent Patterns. In: Fifth IEEE International Conference on Data Mining, pp. 274–281. IEEE Computer Society Press, Los Alamitos (2005)
13. Thomas, S., Bodagala, S., Alsabti, K., Ranka, S.: An Efficient Algorithm for the Incremental Updation of Association Rules in Large Databases. In: Third International Conference on Knowledge Discovery and Data Mining, KDD 1997, Newport Beach, pp. 263–266 (1997)
14. Lee, C.H., Lin, C.R., Chen, M.S.: Sliding-window Filtering: An Efficient Algorithm for Incremental Mining. In: Tenth International Conference on Information and Knowledge Management, CIKM 2001, Atlanta, pp. 263–270 (2001)
15. Cheung, W., Zaiane, O.R.: Incremental Mining of Frequent Patterns Without Candidate Generation or Support Constraint. In: Seventh International Database Engineering and Applications Symposium, pp. 111–116. IEEE Computer Society Press, Los Alamitos (2003)
16. Koh, J.-L., Shieh, S.-F.: An efficient approach for maintaining association rules based on adjusting FP-tree structures1. In: Lee, Y., Li, J., Whang, K.-Y., Lee, D. (eds.) DASFAA 2004. LNCS, vol. 2973, pp. 417–424. Springer, Heidelberg (2004)
17. Frequent Itemset Mining Dataset Repository, `http://fimi.cs.helsinki.fi/data/`

An Efficient Algorithm for Maintaining Frequent Closed Itemsets over Data Stream

Show-Jane Yen, Yue-Shi Lee, Cheng-Wei Wu, and Chin-Lin Lin

Department of Computer Science and Information Engineering, Ming Chuan University,
5 De Ming Rd., Gwei Shan District, Taoyuan County 333, Taiwan
{sjyen,leeys}@mail.mcu.edu.tw

Abstract. Data mining refers to the process of revealing unknown and potentially useful information from a large database. Frequent itemsets mining is one of the foundational problems in data mining, which is to discover the set of products that purchased frequently together by customers from a transaction database. However, there may be a large number of patterns generated from database, and many of them are redundant. Frequent closed itemset is a well-known condensed representation of frequent itemset, and it provides complete information of frequent itemsets. Extensive studies have been proposed for mining frequent closed itemsets from transaction database, but most of them do not take streaming data into consideration. In this paper, we propose an efficient algorithm for maintaining frequent closed itemsets over data streams. Whenever a transaction is added to database, our approach incrementally updates the information of closed itemsets and outputs updated frequent closed itemsets based on user-specified thresholds. The experimental results show that our approach outperforms previous studies.

Keywords: Data Mining, Data Stream, Frequent Closed Itemsets, Frequent Itemsets.

1 Introduction

Data mining [1, 3] refers to the process of revealing unknown and potentially useful information from a large database. *Mining frequent itemsets* from a transaction database is a fundamental task to several data mining applications. The problem of discovering frequent itemsets within database is stated as follows: Given a large database *D* which containing *N* transactions, where each transaction $t \in D$. The finite set of items $I = \{a_1,a_2...,a_M\}$, each transaction *t* is a list of distinct items $\{i_1,i_2,...,i_m\}$, $i_j \in I$, $1 \leq j \leq m$. A *k-itemset* is a set of *k* distinct items $\{i_1,i_2,...,i_k\}$, $i_j \in I$, $1 \leq j \leq k$, where *k* is *the length of itemset.* Given an *itemset X*, *SC*(*X*) is the *support count of X* and defined as the number of transactions in *D* that include *X*. The *support of X* is defined as *SC*(*X*)/*N*. An itemset is said to be *frequent* if its support is no less than a user-specified threshold *min_sup (count)* which is called *minimum support (count).* Otherwise the itemset is said to be *infrequent*. Mining all frequent itemsets from *D* is equivalent to the discovery of all itemsets having a support no less than *min_sup (count).* Let Table 1 be the database, and set *min_sup* = 40%. The *minimum support*

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 767–776, 2009.

count is 40% * 5 = 2.Then, the frequent itemsets are {A}:4, {B}:3, {C}:4, {D}:2, {AB}:3, {AC}:3, {BC}:2, {CD}:2 and {ABC}:2. The number beside each frequent itemset is its support count.

Table 1. A Transaction Database

TID	Transaction
t_1	C,D
t_2	A,B
t_3	A,B,C
t_4	A,B,C
t_5	A,C,D

Many studies [1, 3] for mining frequent itemsets mainly focused on traditional database. They do not take streaming data into consideration. A data stream is an order sequence of transactions that arrives in a timely order. Mining frequent itemsets over data streams is a fundamental problem in several applications such as network traffic analysis, on-line transaction analysis and other important tasks. However, mining in data streams have posed new challenges. First, data come continuously, unbounded and usually with high speed. To keep all information about itemsets from an entire stream is hard. Second, data distribution in streams usually changes with time. The status of itemsets may be changed (from frequent into infrequent or from infrequent into frequent). Third, most analysts are interested in the most recent patterns. They may expect that the information of current frequent itemsets can be output in real time based on user specified-threshold. Some approaches [2, 4] are proposed for above problems. But they often generate a huge set of frequent itemsets and degrade the mining performance.

One of the solutions to this problem is to mine only the frequent closed itemsets [3]. Mining closed itemset is equivalent to mine only those itemsets having no proper supersets with the same support. Moreover, a closed itemset is said to be frequent if its support is no less than *min_sup*. For the above example, the frequent closed itemsets are {A}:4, {C}:4, {AB}:3, {AC}:3 and {ABC}:2. In this case, {B} is non-closed since its support count is same as its superset {AB}. It implicitly indicates that {B} will not appear in a transaction without {A}. Therefore, non-closed itemsets are regarded as redundant.

In general, the number of frequent closed itemsets is much smaller than the number of frequent itemsets. Therefore, mining only the closed itemsets reduces the mining time and memory space. Besides, complete set of frequent closed itemsets can uniquely determine all frequent itemsets and their exactly support without any information loss [6]. Moreover, frequent closed itemset has been proven more meaningful for analysis [5]. Some approaches [2, 4, 6, 7, 8] are proposed for traditional database, but they do not take streaming data into consideration.

Chi et al. [2] propose an algorithm called *Moment* for mining frequent closed itemsets over data streams. It uses a *CET Tree (Closed Enumerate Tree)* to maintain the main information of itemsets. Each node in CET Tree represents an itemset with different node type. Some nodes in CET Tree are not closed so that there are still some redundant nodes in CET Tree. *Moment* must maintain huge CET nodes for a frequent closed itemset. Chi et al. indicated that the ratio of CET nodes for a closed itemsets is about 20:1. If there are a large number of frequent closed itemsets, it will consume a lot of memory space. When a new transaction arrives, the node is inserted and updated according to its node type. The exploration of frequent itemsets and node type checking are time consuming.

CFI-Stream is another algorithm for this problem [4]. Only the closed itemsets are maintained in a lexicographical ordered tree which is called *DIU Tree (DIrect Update Tree).* Each node consists of a closed itemset and its support count. When a new transaction X arrives, CFI-Stream will generate all the subsets of X, and check if each subset Y is closed or not after the transaction arrives. To check whether an itemset Y is closed or not, CFI-Stream may need to search all supersets of Y from *DIU Tree*. It takes a lot of time to generate all the subsets of a new transaction and search their supersets from DIU Tree.

In this paper, we propose an efficient algorithm, called *CloStream*, for maintaining frequent closed itemsets in a data stream. A *Closed Table* and a *Cid List* are used to maintain the information of closed itemsets. All closed itemsets and their support counts are records in *Closed Table*. Each closed itemset has a unique closed itemset identifier, call *cid*. The cids of super closed itemsets for each item are maintained in the Cid List. As a transaction arrives to the database, it incrementally updates closed itemsets and their supports based on previous mining results. Unlike CFI-Stream, CloStream doesn't need to generate subsets of the transaction, and doesn't need to search supersets for each subset. Via the simple intersection of the transaction and certain closed itemsets once. The updated closed itemsets can obtained without multiple scans of whole search spaces. Hence, our approach has better performance than previous approaches.

2 Preliminaries

In this section, we formally define the closed itemsets [6] and describe some properties.

Definition 1 (Closure Operator)

Let T be the subsets of all that transactions in D, $T \subseteq D$, and Y be the subsets of all items appear in D, $Y \subseteq I$. The concept of closed itemset is based on the following two functions f and g:

$$f(T) = \{i \in I \mid \forall t \in T, i \in t\} \quad (1)$$

Function f takes a set of transactions T as an input and returns an itemset included in all transactions belonging to T.

$$g(Y) = \{t \in D \mid \forall i \in Y, i \in t\} \quad (2)$$

Function g takes an itemset Y as an input and returns a set of transactions including Y. A function $C = f \circ g$ is composted by f and g, and is called closure operator [6].

Definition 2 (Closed Itemset)
An itemset X is called a closed itemset if and only if

$$C(X) = f_{\circ} \ g(X) = f(g(X)) = X \qquad (3)$$

$C(X)$ is called *the closure of* X. Definition 2 shows that an itemset X is called *closed* if and only if X equals to its *closure* $C(X)$. Otherwise, X is *non-closed*.

For above example, $g(\{B\}) = \{t_2, t_3, t_4\}$ since these transactions are including {B}. Let $T = \{t_2, t_3, t_4\}$, $f(T) = \{AB\}$ since {AB} belongs to each transaction in T. {B} is non-closed since $C(\{B\}) = \{AB\}$. From above discussions, we have the following definitions and properties.

Definition 3 (Frequent Closed Itemset)
An itemset X is called a *frequent closed itemset* if and only if $X = C(X)$ and its support is no less than *min_sup*.

Property 1. If $Y = C(X)$, then SC(X) = SC(Y) [6].

Property 2. If Y = SC(X), then Y is the smallest closed itemset containing X, $X \subseteq Y$. Moreover, SC(Y) > SC(Z), Z is any superset of X, $X \subset Z$ and $Z \neq Y$.

Reason. Since $X \subset Z$, SC(X) $\geq$ SC(Z). If SC(X) = SC(Z), then each transaction containing X is containing Z, the closure of X is Z instead of Y. It is a contradiction with $Y = C(X)$, $Z \neq Y$. Otherwise, if SC(X) > SC(Z), then SC(Y) > SC(Z) since SC(X) = SC(Y) (Property 1).

Property 3. If SC(X) > SC(Y), Y is any superset of X, $Y \supset X$, then X = C(X) [6].

3 CloStream Algorithm

In this section, we present our proposed CloStream algorithm and its structure. Our approach uses two in-memory data structures which are called *Closed Table* and Cid List respectively. An additional hash table is used to put those itemsets need to be updated as transaction arrives.

Closed Table is used to maintain the information of closed itemsets. Each record of Closed Table represents the information of a closed itemset. It consists of three fields: *Cid*, *CI* and *SC*. Each closed itemset was assigned a unique closed identifier, called *cid*. Cid field is utilized to identify closed itemsets. Given a cid, CloStream gets corresponding closed itemsets in CI field. The support counts are stores in SC field. Initially, the value of first record in Closed Table is set to 0. It will be used in our approach. Table 2 illustrates a Closed Table after five transactions (in Table 1) arrives.

Cid List is used to maintain the items and their *cidsets*. It consists of two fields: *Item* field and *cidset* field. The *cidset of an item X* is denoted as *cidset(X)*, it is a set which contains all cids of *X*'s super closed itemsets. The update of Cid List is stated as follows: As CloStream finds a new closed itemset *Y* and assigns *c* as its cid. Then, *c* will be added into the cidsets of items that are contained by *Y*. For example, if {AB} is closed, whose cid is assigned as 2, then 2 will be added into cidset(A) and cidset(B) respectively. Table 3 illustrates a Cid List. It maintains the items and their superset cids (in Table 2).

A hash table, called $Temp_A$, is used to put those itemsets need to be updated as transaction arrives. It takes *TI* field as key, and *Closure_Id* field as value. Itemsets *S* need to be updated are put in TI field. The closure id of *S* are stored in Closure_Id field. Temp_A was shown in Table 4. In Table 4, the second record is ({B}, 2). The corresponding closed itemset is {AB} in Closed Table (in Table 2). The support count of {B} is same as its closure {AB}.

Table 2. Closed Table after adding t_1, t_2 ,t_3, t_4, t_5

Cid	CI	SC
0	{0}	0
1	{CD}	2
2	{AB}	3
3	{ABC}	2
4	{C}	4
5	{ACD}	1
6	{A}	4
7	{AC}	3

Table 3. Cid List after adding t_1, t_2 ,t_3, t_4, t_5

Item	Cidset
A	{2, 3, 5, 6, 7}
B	{2, 3}
C	{1, 3, 4, 5, 7}
D	{1, 5}

Table 4. The status of Temp_A as t_6 arrives

TI	Closure_Id
{C}	4
{B}	2
{BC}	3

3.1 Maintenance Rules

In this subsection, we discuss the maintenance rules for CloStream. In the following, we assume that t_A is a newly arrived transaction. D_B is the database before adding t_A. $D_U = D_B \cup \{t_A\}$ is an updated database after adding t_A. $SC_{DB}(X)$ and $SC_{DU}(X)$ represent the support counts of itemset *X* within D_B and D_U respectively. $C_{DB}(X)$ and $C_{DU}(X)$ represent the closure of *X* within D_B and D_U respectively. The set of closed itemsets in D_B and D_U are denoted as C_{DB} and C_{DU} respectively. The maintenance rules are based on the following lemmas.

Lemma 1. As t_A arrives to D_B, then itemset $t_A \in C_{DU}$.

Lemma 2. As t_A arrives to D_B, if an itemset *Y* is not a subset of t_A, then the status of *Y* will not be changed, i.e., $SC_{DB}(Y) = SC_{DU}(Y)$ and $C_{DB}(Y) = C_{DU}(Y)$.

Lemma 3. Suppose that an itemset $S = t_A \cap X$, $S \in C_{DB}$. If $S \neq \varnothing$, then *S* is a closed itemset in D_U.

3.2 Adding a Transaction

As a transaction with k items, $t_A = \{i_1, i_2,..., i_k\}$, $i_j \in I$, $1 \leq j \leq k$, is added to the database D_B. CloStream consists of two phases. In first phase, CloStream finds all itemsets need to be updated with their closures in D_B, and puts them into Temp$_A$. In second phase, CloStream updates their support counts according to Property 2, and updates Closed Table and Cid List.

Phase 1. As t_A arrives, according to Lemma 1, t_A is a closed in D_U. CloStream puts t_A into Temp$_A$ and sets its Closure_Id as 0. Since the closure of t_A is unknown at the beginning. The values of Closure_Id will be updated in the middle of mining process. Then, CloStream intersect t_A with associated closed itemsets to get itemsets need to be updated. The set of cids of associated closed itemsets is defined as $SET(\{t_A\})$ = $cidset(i_1) \cup ... \cup cidset(i_k)$. CloStream finds itemsets need to be updated by intersection of t_A and closed itemsets whose cids are in the $SET(\{t_A\})$. According to Lemma 3, the results of intersection are closed itemsets in D_U. This process can be performed by Cid List and Closed Table. Suppose that S is the intersection result of t_A and a closed itemset C which cid $i \in SET(\{t_A\})$. If S is not in Temp$_A$, then put (S, i) into Temp$_A$. Otherwise, if S is already in Temp$_A$ with its current Closure_Id t, compare $SC_{DB}(C)$ with a closed itemset Q which cid is t in Closed Table. If $SC_{DB}(C)$ is greater than $SC_{DB}(Q)$, then CloStream replaces (S, t) which is in Temp$_A$ with (S, i). The reason is that the closure of S has a support greater than any its superset's support (Property 2, 3). The intersections of t_A with C stops till all cids in $SET(\{t_A\})$ are processed. The purpose of phase 1 is to find itemsets need to be updated and find their closure before t_A arrives.

Table 5. Closed Table after adding t_6

Cid	CItemset	SC
0	0	0
1	{CD}	2
2	{AB}	3
3	{ABC}	2
4	{C}	5
5	{ACD}	1
6	{A}	4
7	{AC}	3
8	{B}	4
9	{BC}	3

Table 6. Cid List after adding t_6

Item	Cidset
A	{2, 3, 5, 6, 7}
B	{2, 3, 8, 9}
C	{1, 3, 4, 5, 7, 9}
D	{1, 5}

Phase 2. CloStream gets itemsets X with their Closure_Id c from Temp$_A$, and checks that whether X is equal to closed itemset whose cid is c in Closed Table. If X is already in Closed Table with cid c, then X is originally a closed in D_B. In this case, directly increase $SC_{DU}(X)$ by 1. Otherwise, X is a new closed itemset after t_A arrives. In this case, $SC_{DU}(X)$ is equal to the support count of its closure increased by 1. At the same time, CloStream assigns X a new cid n, puts X into Closed Table, and update Cid List. The phase 2 is completed till all records in Temp$_A$ are processed. Finally, a new set of closed itemsets after t_A arrives can be obtained in new Closed Table. All frequent closed itemsets can be output by scanning Closed Table once.

```
Procedure CloStream ($t_A$, CT, CL)
  $Temp_A$ ← ($t_A$, 0)
  SET({$t_A$}) = cidset($i_1$) ∪ ... ∪ cidset($i_k$)
  for each cid $i$ ∈ SET({$t_A$}) do
        $S$ ← NULL
        $S$ ← $t_A$ •CT[$i$].CI
        if ($S$ ∈ $Temp_A$ ) then
              if (CT[$i$].SC > CT[$t$].SC) then
                   replace ($S$, $i$) with ($S$, $t$) in $Temp_A$
        else
             $Temp_A$ ← $Temp_A$ ∪ (S, i)
  end for
  for each ($X$, $c$) ∈ $Temp_A$ do
      if ($X$ == CT[$c$].CI ) then CT[$c$].SC++
      else
            j ← j+1
            CT ← CT ∪ ( $j$, $X$, CT[$c$].SC+1)
            for each $i_i$ ∈ $t_A$ do cidset($i_i$) ← cidset($i_i$) ∪ j
      end if-else
  end for
end Procedure CloStream
```

3.3 A Running Example for CloStream

Let Table 1 be a running example. Before t_1 arrives, $D_B = \varnothing$. The first record of Closed Table is set to (0, 0, 0). Each cidsets in Cid List is set to $\varnothing$. As t_1 = {CD} arrives, $D_U = D_B \cup t_1$. CloStream puts {CD} into Temp_A and sets its Closure_Id to 0. Then, CloStream unions *cidset*(C) and *cidset*(D) to get *SET*({CD}), i.e., *SET*({CD}) = *cidset*(C)∪*cidset*(D) = $\varnothing$. Since *SET*({CD}) is empty, t_1 does not need to intersect with any closed itemsets. Therefore phase 1 was completed, CloStream goes to phase 2. In phase 2, CloStream updates itemsets within Temp_A by their Closure_Id. Only ({CD}, 0) in Temp_A. CloStream finds a closed itemset whose cid is 0 from Closed Table, *CT*[0] ={0}. Because {0} is not equal to {CD}. {CD} is a new closed itemset after t_1 arrives. Then, CloStream assigns {CD} a new cid which is 1. Then, CloStream determines SC_{DU}(CD), which equals to SC_{DB}(*CT*[0]) increased by 1. Therefore, SC_{DU}(CD) is 1. Finally, CloStream updates Closed Table and Cid List respectively. CloStream inserts (1, {CD}, 1) into Closed Table and inserts 1 into Cid List. Deal with t_2, t_3, t_4 and t_5 in same manners. After transactions in Table 1 are inserted, updated Closed Table and Cid List are shown as Table 3 and Table 4 respectively.

As t_6 = {BC} arrives, CloStream puts {BC} into Temp_A, and sets its Closure_Id to 0. Then, *SET*({BC}) = {2, 3}∪ {1, 3, 4, 5, 7} = {1, 2, 3, 4, 5, 7} (in Table 4). CloStream intersects t_6 with those closed itemsets whose cids belongs to *SET*({BC}). The first is *CT*[1] = {CD} and {BC}∩{CD} = {C}. Put {C} into Temp_A, and set its Closure_Id as 1. Deal with next itemset *CT*[2] = {AB}, i.e., {BC}∩{AB} = {B}. Put ({B}, 2) into Temp_A. Deal with *CT*[3] = {ABC}, i.e., {BC}∩{ABC}={BC}. However, {BC} is already in Temp_A, its current Closure_Id is 0. In this case, CloStream compares SC_{DB}(*CT*[0]) with SC_{DB}(*CT*[3]). As a consequence, SC_{DB}(*CT*[3]) is greater than $SC_{DB}(CT[0])$. According to property 4, CloStream replaces ({BC}, 0) with ({BC}, 3). Deal with rest closed itemsets in same steps. The result of Temp_A was shown in Figure 4.

In phase 2, CloStream updates the support counts of itemsets which are in Temp_A. The first record in Temp_A is ({C}, 4). CloStream finds corresponding closed itemset from Closed Table, which is $CT[4]$ = {C}, and increases SC_{DB}({C}) by 1. Then, deal with next record in Temp_A, which is ({B}, 2). However, {B} is not in original Closed Table. CloStream assigns 8 as its cid. Then, set SC_{DU}({B}) equals to SC_{DB}({AB}) increased by 1. CloStream deals with last record ({BC},3) in same manner. Finally, updated Closed Table and Cid List are shown as Table 6 and Table 7 respectively. A new set of closed itemsets after t_6 arrives are maintained in the updated Closed Table. All frequent closed itemsets can be found by scanning Closed Table once.

4 Experimental Results

In this section, we compare the performance of CloStream against CFI-Stream [4]. These two algorithms are coded in Java language. All experiments are evaluated on a 1.83 GHz Intel Core 2 Duo Processor with 2 Gigabyte memory, and running on Windows Vista. For performance evaluation, we generate the synthetic dataset by IBM data generator [1], publicly available from IBM Almaden.

Table 7. Parameters of IBM data generator

D	The total number of transactions.
T	The average transaction size.
I	The average maximal potential frequent itemset size.
N	Number of distinct items.

Table 7 describes the parameters of synthetic datasets. Two synthetic datasets T5I2D20K and T5I4D20K are generated with fixed N = 2K (K=1000) and T = 5. In order to simulate the environment of a data stream, transactions are inserted to database one by one in our experiments.

Figure 1 shows the running time and memory requirement respectively when performing addition operation. We take 10K transactions as an original database, and evaluate each execution time and memory usage for adding 1K transactions to updated database respectively.i The addition performance of CloStream outperforms than CFI-Stream. The reason is that CFI-Stream needs to check the status of each itemset in new transaction, and needs to find its supersets from DIU Tree. Assume that a transaction with *m* items arrives to database, and CFI-Stream needs to search DIU Tree 2^m-1 times. Different from CFI-Stream, CloStream finds associated closed itemsets by a Cid List, and simply intersect them with new transaction to get closed itemsets need to be updated. Hence, CloStream achieves better performance than CFI-Stream.

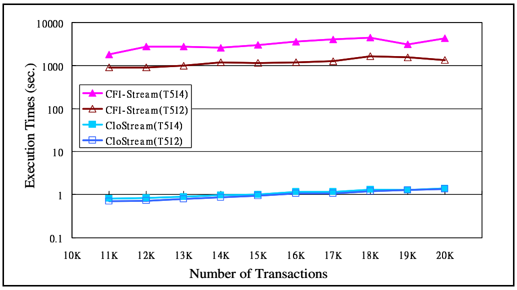


Fig. 1. Execution times after adding transactions

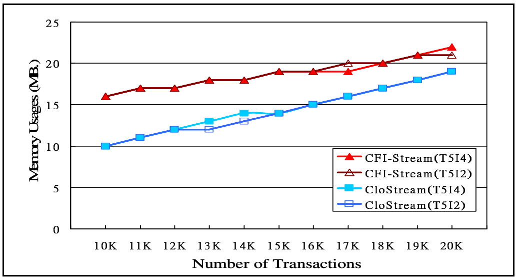


Fig. 2. Memory usages after adding transactions

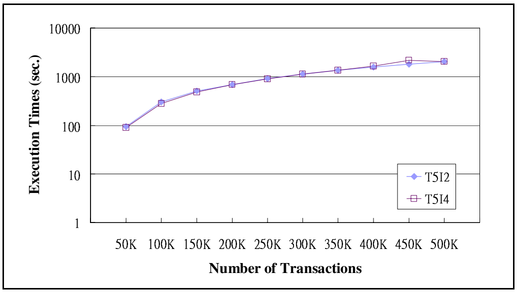


Fig. 3. Scalability on transaction size

Figure 2 indicates that the memory usage of CloStream is lower than CFI-Stream. The reason is that a hash table is attached to improve the execution time of CFI-Stream, and causes higher memory consumption. If CFI-Stream does not attach a hash table, it turns very slow and can not be competed with CloStream.

Figure 3 shows scalability of CloStream. It takes 0K transactions as an original database, and evaluate each execution time for adding 50K new transactions to updated database. Figure 3 only shows the scalability of CloStream, since CFI-Stream is very slow and can not be competed with CloStream. When dataset become larger, CFI-Stream turns to very slow. As shown in Figure 3, CloStream has a good scalability with larger transaction size. Its execution time is increased stably with larger datasets.

5 Conclusions

In this paper, we propose an efficient algorithm, called *CloStream*, for maintaining frequent closed itemsets in data stream. When a transaction arrives to the database, CloStream incrementally updates the closed itemsets without scanning original database. Unlike CFI-Stream [4], CloStream does not need to generate all subsets of the new transaction does not need to search supersets for each subset to determine whether the subset is a closed or not. Extensive experiments are performed to evaluate the efficiency of CloStream and CFI-Stream. In the experiments, CloStream outperforms CFI-Stream over 1000 times, and has good scalability with larger database.

References

1. Agrawal, R., Srikant, R.: Fast Algorithm for Mining Association Rules. In: Proceedings of International Conference on Very Large Data Bases, Santiago, Chile, pp. 487–499 (1994)
2. Chi, Y., Wang, H., Yu, P.S., Muntz, R.R.: Moment: Maintaining Closed Frequent Itemsets over a Stream Sliding Window. In: Proceedings of 2004 IEEE International Conference on Data Mining, Brighton, pp. 59–66 (2004)
3. Han, J., Pei, J., Yin, Y.: Mining Frequent Patterns without Candidate Generation. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, USA, pp. 1–12 (2000)
4. Jiang, N., Gruenwald, L.: CFI-Stream: Mining Closed Frequent Itemsets in Data Streams. In: Proceedings of 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, pp. 592–597 (2006)
5. Ji, L., Tan, K.-L., Tung, A.: Compressed Hierarchical Mining of Frequent Closed Patterns from Dense Data Sets. IEEE Trans. on Knowledge and Data Engineering 19(9), 1175–1187 (2007)
6. Pasquier, N., Bastide, T., Taouil, R., Lakhal, L.: Discovering Frequent Closed Itemsets for Association Rules. In: Proceedings of the 7th International Conference on Database Theory, Jerusalem, Israel, pp. 398–416 (1999)
7. Pei, J., Han, J., Mao, R.: Closet: An Efficient Algorithm for Mining Frequent Closed Itemsets. In: Proceedings of the ACM SIGKDD International Workshop on Research Issues in Data Mining and Knowledge Discovery, Dallas, USA, pp. 21–31 (2000)
8. Wang, J., Han, J., Pei, J.: Closet+: Searching for the Best Strategies for Mining Frequent Closed Itemsets. In: Proceedings of 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, pp. 236–245 (2003)

On Covering Based Approximations of Classifications of Sets

B.K. Tripathy[1] and G.K. Panda[2]

[1] Professor, SCS, VIT University, Vellore – 632014, Tamilnadu, India
[2] Director, MITS School of Biotechnology, Bhubaneswar-760024, Orissa, India
tripathybk@rediffmail.com, gkpmail@sify.com

Abstract. Approximations of classifications, introduced and studied by Grzymala Busse [4], and approximations of sets are two different issues (Pawlak [9]) and the equivalence classes of approximate classifications can not be arbitrary sets. Busse [4] had established properties of approximations of classifications which were recently extended to necessary and sufficient type theorems by Tripathy et al. [12]. Recently, basic rough sets have been extended to introduce covering based rough sets. In this paper we introduce the notion of covering based approximations of classifications and study their properties.

Keywords: Classifications, rough sets, approximations of classifications, covering based rough sets and covering based approximations of classifications.

1 Introduction

Most of our traditional tools for formal modeling, reasoning and computing are crisp, deterministic and precise in character. Real situations are very often not crisp and deterministic and they cannot be described precisely. For a complete description of a real system often one would require by far more detailed data than a human being could ever recognize simultaneously, process and understand. This observation led to the extension of the concept of crisp sets so as to model imprecise data which can enhance their modeling power. One such approach to capture impreciseness is due to Pawlak [6], who introduced the notion of Rough Sets, which is an excellent tool to capture impreciseness in data. The basic assumption of rough set theory is that human knowledge about a universe depends upon their capability to classify its objects. Classifications of a universe and equivalence relations defined on it are known to be interchangeable notions. So, for mathematical reasons equivalence relations were considered by Pawlak to define rough sets. A rough set is represented by a pair of crisp sets, called the lower approximation comprises of elements, which belong to it definitely and upper approximation, which comprises of elements, which are possibly in the set with respect to the available information.

The basic concept of rough sets has been extended in many different directions to extend its modeling power. A covering is a generalization of the notion of partition. Using coverings instead of partitions, covering based rough sets (CB-rough sets) have been introduced by Zakowski [13]. The CB-rough sets are models with promising potential for applications to data mining. There is only one lower approximation for

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 777–786, 2009.

such rough sets. However, there are four different definitions of upper approximation of such sets. Several properties of the different types of CB-rough sets have been derived by different researchers [1,2, 3,13,14,15,16,17,18]. Rough approximation of classifications of sets was introduced and studied by Busse [4]. He has used this concept to study the generation of rules in information systems. Theorems of Busse, which were of sufficient type have been generalized to get necessary and sufficient type results by Tripathy et al. [12], from which several other results have been obtained as corollaries. In the present article we shall extend the results in [12] to the setting of approximation of classifications by coverings. It will be interesting to generate rules from CB-information systems with the help of these results.

2 Definitions and Notations

In this section we introduce some of the definitions and notations, which shall be referred in the rest of the paper.

2.1 Basic Rough Sets

Let U be a universe of discourse and R be an equivalence relation over U. By U/R we denote the family of all equivalence class of R, referred to as categories or concepts of R and the equivalence class of an element $x \in U$, is denoted by $[x]_R$. By a knowledge base, we understand a relational system $K = (U, \Re)$, where U is as above and $\Re$ is a family of equivalence relations over U. For any subset $P\ (\neq \phi) \subset \Re$, the intersection of all equivalence relations in P is denoted by $IND\ (P)$ and is called the indiscernibility relation over P. Given any $X \subseteq U$ and $R \in IND\ (K)$, we associate two subsets, $\underline{R}\, X = \bigcup \{Y \in U/R\colon Y \subseteq X\}$ and $\overline{R}\, X = \bigcup \{Y \in U/R\colon Y \bigcap X \neq \phi\}$, called the R-*lower* and R-*upper approximations* of *X respectively.*

The R-*boundary* of X is denoted by $BN_R\ (X)$ and is given by $BN_R\ (X) = \overline{R}\, X - \underline{R}\, X$. The elements of $\underline{R}\, X$ are those elements of U, which can certainly be classified as elements of X, and the elements of $\overline{R}\, X$ are those elements of U, which can possibly be classified as elements of X, employing knowledge of R. We say that X is *rough* with respect to R if and only if $\underline{R}\, X \neq \overline{R}\, X$, equivalently $BN_R\ (X) \neq \phi$. X is said to be R-*definable* if and only if $\underline{R}\, X = \overline{R}\, X$, or $BN_R\ (X) = \phi$.

2.2 Covering Based Rough Sets

As mentioned in the introduction, there are four types of covering based rough sets introduced in the literature. This is basically due to the four different types of upper approximations introduced. In the next few sections we recall the four types of covering based rough sets [16]. However, we shall be dealing with the first three types only in this paper. However, we shall provide the definitions of all the four types of covering based rough sets here for the sake of completeness.

2.2.1 Covering Based Rough Sets of First Type

We first introduce the concepts necessary for defining the covering based rough sets of first type and define it.

Definition 2.2.1. Let U be a universe of discourse and C be a family of subsets of U. C is called a *cover* of U if no subset in C is empty and ∪C = U. We call (U, C) the *covering approximation space* and the covering C is called the *family of approximation sets.*

Definition 2.2.2. Let (U, C) be an approximation space and x be any element of U. Then the family Md (x) = $\{K\in C\text{: } x \in K \wedge \forall S\in C, (x\in S \wedge S \subseteq K \Rightarrow K= S)\}$ is called the *minimal description of the object* x.

Definition 2.2.3. For any set X⊆U, the family of sets $C_*(X) = \{K\in C\text{: } K \subseteq X\}$ is called the *family of sets bottom approximating the set* X.

Definition 2.2.4. The set $X_* = \bigcup C_*(X)$ is called the CB-*lower approximation of* X.

Definition 2.2.5. The set $X_*^* = X \setminus X_*$ is called the *boundary of the set* X.

Definition 2.2.6. The family of sets Bn(X) = $\bigcup\{Md(x)\text{: } x\in X_*^*\}$ is called the *family of sets approximating the boundary of the set* X.

Definition 2.2.7. The family of sets $C^*(X) = C_*(X) \bigcup Bn(X)$ is called the *family of sets top approximating the set* X.

Definition 2.2.8. The set $X^* = \bigcup C^*(X)$ is called the CB-*upper approximation of the first type of the set* X.

Definition 2.2.9. A set X is called *exact* when $C^*(X) = C_*(X)$.Equivalently, X is exact when $X^* = X_*$. Otherwise, X is said to be a covering based rough set *first type.*

We would like to note here that for all the four types of covering based rough sets, the definition of lower approximation is same.

2.2.2 Covering Based Rough Sets of Second Type

Definition 2.3.1. Let (*U*, *C*) be a covering approximation space. For a set $X \subseteq U$, the set $X_2^* = \bigcup\{$ K/ $K\in C,\ K \bigcap X \neq \phi \}$ is called the *covering upper approximation of second type of* X.

If $X_* = X_2^*$ is said to exact. Otherwise, X is called a *covering based rough set of the second type.*

2.2.3 Covering Based Generalized Rough Sets of Third Type

Definition 2.4.1. Let (*U*, *C*) be a covering approximation space. For a set $X \subseteq U$, the set $X_3^* = \bigcup\{Md(x) | x\in X\}$ is called the *covering based upper approximation of the* *third type of* X . If $X_* = X_3^*$, X is said to be exact. Otherwise, X is called a *covering based rough set of the third type.*

2.2.4 Covering Based generalized Rough Sets of Fourth Type

Definition 2.5.1. Let (U, C) be a covering approximation space. For a set $X \subseteq U$, the set $X_4^* = X \cup \{\ K/\ K \in C\ and\ \ K \cap (X \setminus X_*) \neq \phi\ \}$ is called the *covering based upper approximation of the fourth type of* X . If $X_* = X_4^*$, X is said to be exact. Otherwise, X is called a *covering based rough set of the fourth type.*

3 Approximations of Families of Sets

As mentioned above, the concept of a rough set is defined through two crisp sets, namely the lower and upper approximation of a set. These concepts of approximations of sets have been extended to family of sets also. In this section we shall introduce the existing approximations and also propose a new type of approximation in the form of approximations of covering based rough sets.

3.1 Basic Approximations of Classifications

Classifications of universes play a central role in the study of basic rough set theory. Extending the concept of approximation of sets, Grzymala-Busse [4] introduced the approximation of classifications. As pointed out by Pawlak ([10], p.24) these results of Busse establish that the two concepts, approximation of sets and approximation of families of sets (or classifications) are two different issues and that the equivalence classes of approximate classifications cannot be arbitrary sets. He has further stated that if we have positive examples of each category in the approximate classification then we must also have negative examples of each category. We define below a classification on an universe formally.

Definition 3.1.1. Let F = $\{\mathrm{X}_1, \mathrm{X}_2 \ldots \mathrm{X}_n\}$ be a family of non empty sets defined over U. We say that F is a *classification of* U if and only if $\mathrm{X}_i \cap \mathrm{X}_j = \phi$ for $\mathrm{i} \neq \mathrm{j}$ and $\bigcup_{i=1}^{n} X_i = \mathrm{U}$.

Definition 3.1.2. Let F be as above and R be an equivalence relation over U. Then $\underline{R}$ F and $\overline{R}$ F denote respectively the R-*lower* and R-*upper approximations* of the family F and are defined as

$$\underline{R}\,\mathrm{F}=\{\ \underline{R}\,\mathrm{X}_1, \underline{R}\,\mathrm{X}_2, \ldots., \underline{R}\,\mathrm{X}_n\ \},\ \ \overline{R}\,\mathrm{F}=\{\ \overline{R}\,\mathrm{X}_1,\ \overline{R}\,\mathrm{X}_2,\ \ldots.\ \overline{R}\,\mathrm{X}_n\ \}.$$

Grzymala-Busse [4] has established some properties of approximation of classifications in the form of four theorems. These results are irreversible by nature.

However, these results have been further extended in [12] to obtain theorems establishing necessary and sufficient type of properties of approximation of classifications. From these general types of results, many interesting results have been obtained as corollaries, besides the results of Busse.

3.2 Theorems on Basic Approximation of Classifications

In this section, we shall quote two theorems generalizing the four theorems established by Busse [4] on approximation of classifications and the ten corollaries derived there on by Tripathy et al. [12]. These theorems are of necessary and sufficient type. The interpretation of all these results and their illustration through examples is provided in [12].
We shall use the following notations for representational convenience:

$$N_n = \{1, 2, ..., n\}.$$

For any $I \subset N_n$, I^c is the complement of I in N_n.

In all the following results we take $F = \{X_1, X_2, ..., X_n\}$ as a classification of U and R as an equivalence relation on U.

Theorem 3.2.1. For any $I \subset N_n$, $\bar{R}(\bigcup_{i\in I} X_i) = U$ if and only if $\underline{R}(\bigcup_{j\in I^C} X_j) = \phi$.

Corollary 3.2.1. For any $I \subset N_n$, if $\bar{R}(\bigcup_{i\in I} X_i) = U$ then $\underline{R}X_j = \phi$ for each $j \in I^C$.

Corollary 3.2.2. For each $i \in N_n$, $\bar{R}X_i = U$ if and only if $\underline{R}(\bigcup_{j\neq i} X_j) = \phi$.

Corollary 3.2.3. For each $i \in N_n$, $\underline{R}X_i = \phi$ if and only if $\bar{R}(\bigcup_{j\neq i} X_j) = U$.

Corollary 3.2.4. If there exists $i \in N_n$ such that $\bar{R}X_i = U$ then for each j other than i in N_n, $\underline{R}X_j = \phi$.

Corollary 3.2.5. If for all $i \in N_n$, $\bar{R}X_i = U$ holds then $\underline{R}X_i = \phi$ for all $i \in N_n$.

Theorem 3.2.2. For any $I \subset N_n$, $\underline{R}(\bigcup_{i\in I} X_i) \neq \phi$ if and only if $\bigcup_{j\in I^c} \bar{R}X_j \neq U$.

Corollary 3.2.6. For $I \subset N_n$, if $\underline{R}(\bigcup_{i\in I} X_i) \neq \phi$ then $\underline{R}X_j \neq U$ for each $j \in I^C$.

Corollary 3.2.7. For each $i \in N_n$, $\underline{R}X_i \neq \phi$ if and only if $\bigcup_{j\neq i} \bar{R}X_j \neq U$.

Corollary 3.2.8. For all $i, 1 \leq i \leq n$, $\bar{R}X_i \neq U$ if and only if $\underline{R}(\bigcup_{j\neq i} X_j) \neq \phi$.

Corollary 3.2.9. If there exist $i \in N_n$ such that $\underline{R}X_i \neq \phi$ then for each j $(\neq i) \in N_n$, $\underline{R}X_j \neq U$.

Corollary 3.2.10. If for all $i \in N_n$, $\underline{R}X_i \neq \phi$ holds then $\bar{R}X_i \neq U$ for all $i \in N_n$.

4 Covering Based Approximations of Classifications

In this section we shall extend the properties of basic approximations of classifications to the setting of covering based approximations of classifications. It is worth noting that because of the generalized situation, the theorems can not be extended in their full force. However, the results of Busse could be extended to this general setting.

4.1 Approximation of Covering Based Rough Sets

Covering based approximations of classifications is an extension of the basic approximations of classifications, which are to deal with the same lower approximations for all types of covering based approximations and different types of upper approximations for different types.

Let F = {X$_1$, X$_2$… X$_n$} be a classification of U. Then for any other cover C of U we define $F_* = \{X_{1*}, X_{2*}, ..., X_{n*}\} and\ F_i^* = \{X_{1i}^*, X_{2i}^*, ..., X_{ni}^*\}$, called the *C-lower approximation* and *C-upper approximation of type-i* (i=1, 2, 3) of the classification F respectively; where $X_{i*}\ and\ X_{ij}^*\ are$ respectively the C-lower approximation and C-upper approximation of type-j (j= 1, 2, 3) of X_i , i = 1, 2,…n respectively.

We require the following theorems from Zhu and Wang [16] for proving the results of this paper:

Theorem A. For any two covering based rough sets X,Y of any type we have

$$X_* \bigcup Y_* \subseteq (X \bigcup Y)_*. \tag{4.1}$$

Equality holds in (4.1) if and only if

For every $K_1, K_2 \in C, K_1 \bigcap K_2$ is a union of finite elements in C. (4.2)

Theorem B. For any two covering based rough sets X,Y of any type we have

$$X_* \bigcap Y_* \supseteq (X \bigcap Y)_*. \tag{4.3}$$

Theorem C. For any two covering based rough sets X, Y of type-i, we have

$$(X \bigcap Y)_i^* \subseteq X_i^* \bigcap Y_i^*, \text{ i=1, 2, 3.} \tag{4.4}$$

Theorem D. For any two covering based rough sets X, Y of type-i, we have

$$(X \bigcup Y)_i^* \subseteq X_i^* \bigcup Y_i^*, \text{ i=1, 2, 3.} \tag{4.5}$$

Equality holds in (4.5), for i =2 or i = 3 and for i = 1 equality holds if and only if (4.2) is true.

Theorem E. For any two covering based rough sets X, Y of any type we have,

$$X \subseteq Y \Rightarrow X_* \subseteq Y_*. \tag{4.6}$$

Theorem F. For any two covering based rough sets X, Y of type-i, we have

$$(X \subseteq Y \Rightarrow X_i^{*} \subseteq Y_i^{*}) \tag{4.7}$$

for i =2 or i = 3 and for i = 1it holds if and only if (4.2) holds.

Theorem G. Let X be a covering based rough sets of type *i*. Then we have

$$(\sim X)_i^{*} \supseteq \sim (X_*), \quad i = 1,2,3. \tag{4.8}$$

(i) No sufficient condition have been found as for the equality to hold in (4.8) for covering based rough set of type 1.

(ii) In general equality does not hold good for covering based rough set of type 2. However equality holds in this case if and only if C is a partition.

Now we establish the theorems of the paper. In all the results, we take $F = \{X_1, X_2, ..., X_n\}$ as a classification of U and let C as a covering of U.

Theorem 4.1. For any $I \subset N_n$, if $(\bigcup_{j\in I^C} X_j)_* = \phi$ then $(\bigcup_{i\in I} X_i)_j^{*} = U$, j = 1, 2 3.

Proof: We have $(\bigcup_{j\in I^C} X_j)_* = \phi \Rightarrow (U - \bigcup_{j\in I} X_j)_* = \phi$. But, by Theorem G

$$(U - \bigcup_{j\in I} X_j)_* \supseteq -(\bigcup_{i\in I} X_i)_j^{*} .$$

So, $$-(\bigcup_{i\in I} X_i)_j^{*} = \phi .$$

Hence, $$(\bigcup_{i\in I} X_i)_j^{*} = U.$$

This completes the proof.

Corollary 4.1. Let $i \in N_n$ .Then we have if $(X_i)_j^{*} = U$ then $(\bigcup_{j\neq i} X_j)_* = \phi$ for j = 1,2,3

Proof: Taking $I = \{i\}$ in Theorem 4.1 we get this.

As a corollary to corollary 4.1 we get a part Theorem 3.2.1.

Theorem 4.2. Let $I \subset N_n$. Then

(i) $(\bigcup_{i\in I} X_i)_* \neq \phi$ if and only if $\bigcup_{i\in I^C} (X_i)_2^{*} \neq U$.

(ii) $\bigcup_{i\in I^C} (X_i)_3^{*} \neq U$ *if* $(\bigcup_{i\in I} X_i)_* \neq \phi.$

(iii) $\bigcup_{i\in I^C} (X_i)_1^{*} \neq U$ *if* $(\bigcup_{i\in I} X_i)_* \neq \phi$ *and condition* (4.2) *holds.*

For j=1 we require the additional condition (4.2) to hold.

Proof:

(i) (*Sufficiency*) By property of CB-upper approximations, $(\bigcup_{i\in I^C} X_i)_2^* = \bigcup_{i\in I^C} (X_i)_2^* \neq U$.

So, there exists K in C such that, $x \in K$ *and* $K \cap \bigcup_{i\in I} X_i = \phi$. *Hence* $K \subset \bigcup_{i\in I} X_i$. *So*, $x \in \bigcup_{i\in I} X_i$. *That is* $(\bigcup_{i\in I} X_i)_* \neq \phi$.

(*Necessity*) Suppose, $(\bigcup_{i\in I} X_i)_* \neq \phi$ Then there exists $x \in U$ such that $K \subseteq \bigcup_{i\in I} X_i$.

Thus, $K \cap X_k = \phi$ for $k \notin I$. So, $x \notin (X_i)_2^*$, for $i \notin I$.

Hence, $\bigcup_{i\in I^C} (X_i)_2^* \neq U$.

(ii) The proof is similar to the sufficiency part of (i).
(iii) The proof is similar to the sufficiency part of (i). We use Theorem F here.

Corollary 4.2. For $I \subset N_n$, if $(\bigcup_{i\in I} X_i)_j^* \neq \phi$ then $(X_i)_* \neq U$ for each $i \in I^C$ and j = 1, 2, 3.

Proof: By Theorem 4.2, $(\bigcup_{i\in I} X_i)_* \neq \phi$

$$\Rightarrow \bigcup_{i\in I^C} (X_i)_j^* \neq U$$

$$\Rightarrow (X_i)_j^* \neq U \text{ for each } i \in I^C.$$

This completes the proof.

Corollary 4.3. For each $i \in N_n, (X_i)_* \neq \phi$ if and only if $\bigcup_{k\neq i} (X_k)_j^* \neq U$.

Proof: Taking $I = \{i\}$ in Theorem 4.2 we get this.

Corollary 4.4. For all $i, 1 \leq i \leq n$, $(X_i)_j^* \neq U$ if and only if $(\bigcup_{j\neq i} X_j)_* \neq \phi$.

Proof: Taking $I = \{i\}^C$ in Theorem 4.2 we get this. Also, this result can be obtained as a contra positive of Corollary 4.3.

Corollary 4.5. If there exist $i \in N_n$ such that $(X_i)_* \neq \phi$ then for each k $(\neq i) \in N_n, (X_k)_* \neq U$.

Proof: By Corollary 4.3, $(X_i)_* \neq \phi \Rightarrow \bigcup_{k\neq i} (X_k)_j^* \neq U$ for each $k \neq i,\ 1 \leq k \leq n$.

Corollary 4.6. If for all $i \in N_n$, $(X_i)_* \neq \phi$ holds then $(X_i)_j^* \neq U$ for all $i \in N_n$ *and* j=1, 2, 3.

Proof: As $(X_i)_* \neq \phi$ for all $i \in N_n$, we have $(\bigcup_{j \neq i} X_j)_* \neq \phi$ for all $i \in N_n$. So, by Corollary 4.5 $(X_i)_j^* \neq U$ for all $i \in N_n$.

5 Conclusions

In this article we introduced the important concept of covering based approximation of classifications. This concept extends the notions of basic approximations of classifications introduced and studied by Busse [4]. The properties of the basic approximations which were established by Busse in the form of four theorems were extended to two theorems of the necessary and sufficient type by Tripathy et al. [12]. By this many properties other than those of Busse could be obtained. In this paper we tried to extend the theorems of [12], which obviously extend the corresponding results of Busse. However, as expected, the results could not be obtained in their full generality. Sufficient conditions can be obtained for the theorems to hold in their full generality. It is worth noting these results can be used to derive rules from information systems having domain values of attributes inducing covers instead of a partition on the data sets.

References

1. Bonikowski, Z., Bryniarski, E., Wybraniec, U.: Extensions and intensions in the rough set theory. J. Information Sciences 107, 149–167 (1998)
2. Degang, C., Changzhong, W., Qinghua, H.: A new approach to attribute reduction of consistent and inconsistent covering decision systems with covering rough sets. J. Information Sciences 177, 3500–3518 (2007)
3. Degang, C., Wenxiu, Z., Yeung, D., Tsang, E.C.C.: Rough Approximations on a complete completely distributive lattice with applications to generalized rough sets. J. Information Sciences 176, 1829–1848 (2006)
4. Grzymala-Busse, J.: Knowledge acquisition under uncertainty – a rough set approach. J. Intelligent and Robotics Systems 1, 3–16 (1988)
5. Novotny, M., Pawlak, Z.: Black Box Analysis and Rough Top Equality. Bull. Polish Acad. Sci. Math. 33, 105–113 (1985)
6. Novotny, M., Pawlak, Z.: Characterization of Rough Top Equalities and Rough Bottom Equalities. Bulletin Polish Acad. Sci. Math. 33, 91–97 (1985)
7. Novotny, M., Pawlak, Z.: On Rough Equalities. Polish Acad. Sci. Math. 33, 99–104 (1985)
8. Pawlak, Z.: Rough Sets. J. Inf. Comp. Sc. II, 341–356 (1982)
9. Pawlak, Z.: Rough Sets-Theoretical Aspects of Reasoning about Data. Kluwer Acad. Publ., Dordrecht (1991)
10. Pawlak, Z., Skowron, A.: Rough sets some extensions. J. Information Sciences 177(1), 28–40 (2007)
11. Tripathy, B.K., Mohanty, D., Ojha, J.: On Rough Definability and Types of Approximation of Classification. In: IEEE International Conference to be held in Thapper Engg. College, Patiala from March 5-7 (2009)

12. Tripathy, B.K., Mohanty, D., Ojha, J.: On Rough Approximations of Classifications and Representation of Knowledge. In: Communicated to 1st IIM International Conference on Advanced Data Analysis, Business Analytics and Intelligence, June 6-7 (2009)
13. Zakowski, W.: Approximation in space. J. Demonstratio Mathematica 16, 761–769 (1983)
14. Zhu, W., Wang, F.Y.: Reduction and Axiomization of Covering Generalized Rough sets. J. Information Sciences 152, 217–230 (2003)
15. Zhu, W., Wang, F.Y.: Relationships among Three Types of Covering Rough Sets. In: IEEE GrC 2006, pp. 43–48. IEEE Press, Los Alamitos (2006)
16. Zhu, W.: Basic Concepts in Covering-based Rough Sets. In: 3rd IEEE International Conference on Natural Computation. IEEE Computer Society, Los Alamitos (2007)
17. Zhu, W.: Topological Approaches to Covering Rough Sets. J. Information Sciences 177, 1499–1508 (2007)
18. Zhu, W.: On Three Types of Covering-Based Rough Sets. IEEE Transactions on Knowledge and Data Engineering 19(8), 1131–1143 (2007)

NPUST: An Efficient Clustering Algorithm Using Partition Space Technique for Large Databases

Cheng-Fa Tsai and Heng-Fu Yeh

Department of Management Information Systems,
National Pingtung University of Science and Technology,
91201 Pingtung, Taiwan
{cftsai,m9656007}@mail.npust.edu.tw

Abstract. The rapid progress of information technology has led to increasing amounts of data produced and stored in databases. How to extract the implicit and useful information with lower time cost and high correctness is of priority concern in data mining, explaining why many clustering methods have been developed in recent decades. This work presents a new clustering algorithm named NPUST, which is an enhanced version of KIDBSCAN. NPUST is a hybrid density-based approach, which partitions the dataset using K-means, and then clusters the resulting partitions with IDBSCAN. Finally, the closest pairs of clusters are merged until the natural number of clusters of dataset is reached. Experimental results indicate that the proposed algorithm can handle the entire cluster, and efficiently lower the run-time cost. They also reveal that the proposed new clustering algorithm performs better than several existing well-known approaches such as the K-means, DBSCAN, IDBSCAN and KIDBSCAN algorithms. Consequently, the proposed NPUST algorithm is efficient and effective for data clustering.

Keywords: Data mining, data clustering, density-based clustering, partitioning clustering.

1 Introduction

The rapid technological advance implies that the amount of data stored in databases is rising very fast. However, data mining can extract valuable implicit information in large databases. Data clustering is the most widely employed data mining technique for various business applications. Data clustering categorizes objects into groups of high similar. Conversely, objects in different groups are quite dissimilar. Many clustering algorithms have been presented in recent years [1]-[11].

Most clustering algorithms can be classified as partitioning, density-based, hierarchical, grid-based or mixed algorithms. A partitioning clustering algorithm partitions a dataset into k clusters containing similar objects. The best-known partitioning clustering algorithm is K-means [4]. K-means is easily and quickly

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 787–796, 2009.

implemented, and quickly identifies optima for data clustering, but cannot accurately recognize arbitrary shapes. The clustering concept of a dense region is unique for density-based clustering, which can handle arbitrarily shaped clusters and can filter out noise, and produces stable clustering results. However, it consumes a large run-time cost. Typical density-based clustering algorithms are DBSCAN [5] and IDBSCAN [6]. Hierarchical clustering constructs a hierarchical structure from the dataset, and has two clustering procedures for hierarchical structure, namely bottom-up (agglomerate) and top-down (divisive). Well known agglomerate clustering algorithms, which merge similar objects to a cluster, include BIRCH [7], CURE [8], and ROCK [9]. Divisive clustering algorithms, such as CHAMELEON [10], decompose the cluster of difference into numerous clusters. Nevertheless, hierarchical clustering needs to compare all objects before they are merged or decomposed, which is a time-consuming operation. Grid-based clustering adopts grid-cell structure in dataset to cluster. Each grid-cell is treated as a point, thus all clustering procedures are performed on a grid-cell structure. Gird-based clustering has the merit of the fast clustering time, but cannot work efficiently in high-dimensional space. CLIQUE [11] is a classic grid-based clustering algorithm. Mixed clustering algorithms, such as GDH [1], KIDBSCAN [2] and ANGEL [3], can be placed in more than one category.

To decrease the time taken by density-based clustering, this investigation presents a new algorithm **NPUST** (a **N**ew **P**rogram **U**sing narrow **S**earch region **T**echnique based on density-based clustering) by hybridizing partitioning, density-based and hierarchical clustering approaches, an advanced version of KIDBSCAN. This new algorithm is enhanced in two important directions, namely partition space of dataset for searching and efficient agglomeration. Simulation results demonstrate that the proposed NPUST algorithm is a highly effective and efficient clustering method.

The rest of this paper is organized as follows. Section 2 describes several clustering algorithms related to our work. The idea of partition space technique and the proposed algorithm NPUST are presented in Section 3 and 4 respectively. Next, Section 5 shows the simulation results. Conclusion is drawn in Section 6.

2 Related Works

McQueen developed **K-means** in 1967. It is one of most commonly used clustering algorithm in data mining, owing to its efficiency and scalability in clustering large databases. K-means involves the following procedures. (1) Initialize the cluster centers K with random sampling. (2) Assign each object to the nearest cluster center. (3) Compute the new K cluster centers. (4) Repeat steps 2 and 3 until K cluster centers convergence. However, K-means has some weaknesses: it cannot filter out noise; it cannot discover effectively in arbitrary shapes, and its clustering result is instable.

The first density-based clustering algorithm was **DBSCAN**, which takes two input parameters *Eps* and *Minpts*. The main concept of DBSCAN is that a neighborhood around a point of a given radius (*Eps*) must include at least minimum number of points (*Minpts*). DBSCAN begins cluster action from an arbitrary point *p*. The neighborhood of *p* is obtained by executing a region query. If the neighborhood of *p* includes greater than *Minpts* points, then a new cluster with *p* as the *corepoint* is created, and all data points in neighborhood of *p* are assigned to this cluster. Otherwise, the point *p* is labeled as *noise*. The neighborhood of each of *p*'s neighbors is then examined to see whether it can be assigned into the cluster. The process is repeated for every point in this neighborhood. If a cluster cannot be expanded further, then DBSCAN chooses another arbitrary unclassified point, and repeats the process. This procedure is iterated until all points in the dataset have been categorized or labeled as *noise*. Furthermore, if a point is labeled as *seed* and *noise*, then the point is named a *borderpoint*. DBSCAN can locate arbitrary shapes and separate noises. Nevertheless, the complexity of DBSCAN is very high for large databases, since each point must check all data points to find its neighborhood.

IDBSCAN is an advanced version of DBSCAN, which applies sampling-based approach to lower the number of expansion queries owing to the large number of expansion queries of DBSCAN. For neighborhood of each point exceeds *Minpts* points, DBSCAN considers all points in neighborhood as seed to expand cluster, thus generating redundant actions of expansion that query the same neighborhood many times. The technique sets eight distinct points in the neighborhood region (a circle with radius *Eps*) as Marked Boundary Objects (MBOs), and then chooses representative points nearest MBOs as a seed. If the representative point is nearest more than one MBO, then it is selected as a seed only once. Hence, the number of seeds chosen is at most eight for the neighborhood region of each *corepoint*. This scheme decreases the number of expansion queries and execution time dramatically. IDBSCAN still generates the same accurate clustering result of DBSCAN.

IDBSCAN performs data clustering based on the sequence of dataset, making it inefficient due to the time taken to expand a cluster from low-density points or border points. Accordingly, **KIDBSCAN** is presented to solve this problem. KIDBSCAN is a hybrid algorithm that combines K-means and IDBSCAN. K-means is employed to yield high-density center points, which are then moved to the front of data set. Clusters are expanded from high-density center points by performing IDBSCAN. The expansion of clusters from high-density center points then decreases the number of redundant steps and raises efficiency. KIDBSCAN provides high-quality data clustering, and significantly reduces the I/O cost.

3 The Idea of Partition Space Technique

DBSCAN, IDBSCAN, and KIDBSCAN seek clusters by exploring the neighborhood of each unclassified point, and expand a cluster from a core point *P* by conducting neighborhood queries for each *seed* point. However, querying all points in a dataset to find neighborhood of each unclassified and seed point is very expensive. Fig. 1a depicts clustering starting from *p*, where the whole region must be searched to find the neighborhood of *p*. If a dataset contains *n* points,

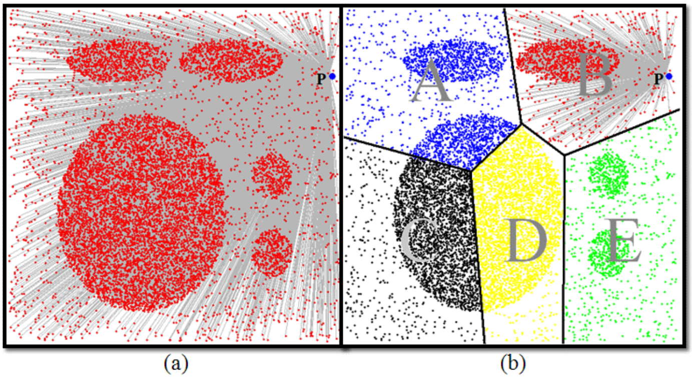


Fig. 1. The concept of searching of neighborhood. (a) The search region covers all points. (b) Region B, in which point p is located.

then each point must query n points. The number of queries is n^2. Thus, the query time is long.

Reducing the query time and number of queries is an important issue. This work presents a partition space technique that adopts dataset partitioning to lower the search zone of the clustering technique. The dataset is partitioned fast into K clusters by K-means. Fig. 1b illustrates a dataset partitioned into five clusters as A, B, C, D and E. Points in region B, where point p is located, is scanned to discover the neighborhood of p. Therefore, the number of queries becomes nk, where k denotes data points number of cluster.

4 The Proposed NPUST Clustering Algorithm

This section describes the proposed NPUST clustering algorithm in detail, and presents an illustrative example to explain the clustering process. This algorithm can be described in terms of the following three stages.

(1) **Partitioning the dataset:** This step decreases the number of queries. DB-SCAN, IDBSCAN and KIDBSCAN search for *corepoint* to extract dense regions and form a new cluster. Checking each point for core conditions means searching the entire dataset to obtain the neighborhoods within the specified *Eps* threshold, and is thus a computationally expensive process. However, partitioning the dataset can enhance clustering efficiency, where the neighborhood search is applied on the whole partition only. It reduces the number of dataset queries by restricting the search space for each unclassified or *seed* point to the partitions, rather than to the whole dataset.

K-means is one of the most efficient partitioning techniques for very large datasets, according to the related work introduced in the previous sections. The proposed algorithm utilizes K-means as a preprocessing stage, and IDBSCAN adopts the resulting partitions to detect their dense regions separately. The dataset partitions yielded by K-means depend on one parameter, which is the natural number of clusters of dataset.

(2) **Clustering each partition using IDBSCAN:** IDSBCAN identifies clusters of dataset for each partition separately to improve the performance and lower the number of scans. Since K-means is a centroid-based scheme, its output partitions may contain points from other clusters that occur as a result of assigning each point to its nearest centroid. This case generally occurs in arbitrary shapes, and even in spherical shapes with different sizes. Under the specified IDBSCAN parameters *Eps* and *MinPts*, these points are not reachable from others within the same partition. Depending on the accepted solution from the partitioning stage, IDBSCAN can detect more than one dense region in each partition, where the number of dense regions is larger than the final natural number of clusters that can be detected by applying IDBSCAN directly onto the dataset. This introduces the need for a new stage to merge the nearest dense regions detected by IDBSCAN from the different partitions to reach the final natural number of clusters. Furthermore, IDBSCAN filters out the noise from each dense region.

(3) **Merging closest clusters:** A merging stage is stipulated to merge the nearest dense regions detected by applying IDBSCAN separately on each partition in order to reach the final number of clusters. Single linkage is employed as the basis for combining dense regions from various clusters. The single linkage means the shortest distance between the closest pair of data points in two different dense regions. This approach is not efficient if all objects in a dense region are applied to measure a single linkage between pairs of dense regions. Hence, border points are used instead of using all dense region objects for this purpose. Border points are extracted automatically when running IDBSCAN, and are defined in the section of related work. As the dense regions are merged, the final natural number of clusters is reached exactly as the output clusters detected by applying the IDBSCAN directly on the whole dataset. Two dense regions i and j are merged if they conform to the following equation:

$$d_{min}(C_i, C_j) = min_{p \in C_i, p' \in C_j} \parallel P - P^{'} \parallel \tag{1}$$

Where C_i and C_j represent two difference clusters referring to two dense regions. P and $P^{'}$ denote the data points in C_i and C_j respectively.

An illustrative example is then presented to explain the clustering process below.

Fig.2a shows DataSet-3 [3], which contains 115,000 points in 2 dimensional space representing 5 clusters with spherical shapes. The first stage is to partition the dataset using K-means. In this example, the required number of partitions is set to 5. Because K-means is a centroid-based approach, it fails to detect the final solution, and splits some clusters into different partitions, where the points are assigned to their closest centroid as displayed in Fig. 2b. Fig. 2c presents the 7 dense regions, which exceeds the natural number of 5 clusters, obtained by applying IDBSCAN separately on each partition. Border objects are detected automatically during the run of IDBSCAN as described in Fig. 2d. Clusters are merged until the number of clusters equals the required number of partitions in Fig. 2e according to the border points of the dense regions as Eqn. (1) shows.

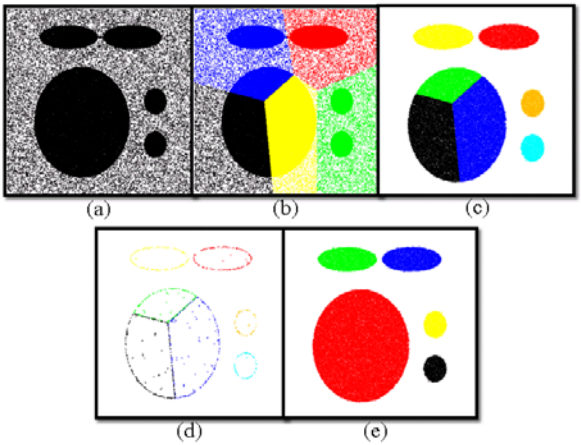


Fig. 2. Clusters obtained by NPUST

The NPUST clustering algorithm can be described as follows:

```
NPUST(DataSets, K, Eps, Minpts)
    Initialization;
    ExecuteK_means(DataSets, K);
    PartitionsStorage();
    ExecuteIDBSCANonPartition(PartitionData, Eps, minpts)
    WHILE ParitionNumber > K then
      MergeClosestPartition(PartitionData);
    End WHILE
End NPUST
```

`DataSets` indicates an input dataset. `K` represents the required number of partitions of dataset, `Eps` denotes the radius of neighborhood of each point and `Minpts` is the minimum number of points in the radius of neighborhood of each point. The algorithm is presented step by step below.

1. Initialization of all arguments.
2. Perform Execute `K_means()` function to partition the space of dataset into K partitions.
3. Execute `PartitionStorage()` function to store the data of partitions.
4. Conduct `ExecuteIDBSCANonPartition()` function to recognize each partition by IDBSCAN.
5. Repeat the process by while loop.
6. Perform `MergeClosestPartition()` function to merge the nearest pairs of partition and then yield accurate clusters.

Clusters are merged until the number of clusters equals the required number of partitions (K).

5 Experiment and Analysis

Experiments were conducted on a desktop computer with 2GB RAM and an Intel 3.4GHz Dual-CPU and running Microsoft Windows XP Professional. All

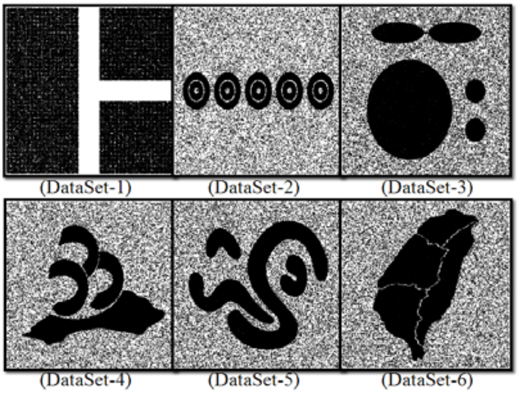


Fig. 3. The original datasets

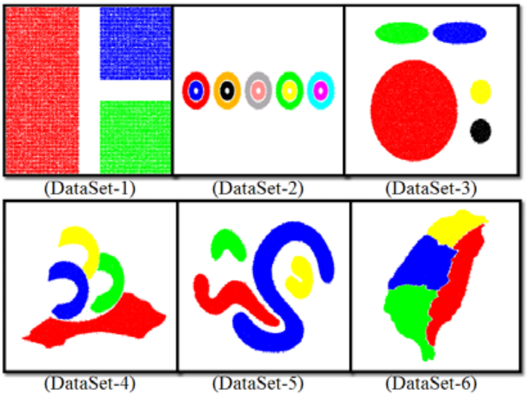


Fig. 4. The results of clustering by NPUST

algorithms were implemented in a Java-based program. This simulation employed six synthetic datasets. The DataSet 1, containing 100,000 points without noise in 2-D, was obtained from Tsai [2]. The patterns of datasets 2, 3, 4, 5 and 6 were sampled Tsai [3]. These datasets have three data sizes with 115,000, 230,000 and 575,000 points in 2-D, and with 15% noise. Fig. 3 illustrates the original datasets. For DataSet 1, the parameters K, *Eps* and $Minpts$ were set to 3, 5 and 35. For DataSets 2-6, the *Eps* all were set to 6; K were set to 10, 5, 4, 4, and 4, respectively, and $Minpts$ were set to 95, 40, 50, 50, and 45, respectively. The results of the proposed NPUST algorithm were compared with those of K-means, DBSCAN, IDBSCAN and KIDBSCAN. For clustering performance comparisons, the clustering correctness rate (CCR) and noise filtering rate (NFR) are shown, where CCR represents the percentage of cluster points correctly recognized by algorithm, and NFR denotes the percentage of noise points correctly filtered by algorithm. Owing to the limitation of length, not all experimental results

Table 1. Comparisons of NPUST, K-means, DBSCAN, IDBSCAN and KIDBSCAN using 115,000 datasets with 15% noise with DataSets 2-6; the DataSet 1 has 100,000 points without noise; item 1 represents run-time cost (in seconds); item 2 denotes CCR (%), while item 3 indicates NFR (%)(N/A means that the simluations were not performed)

Algorithm	Item	DataSet-1	DataSet-2	DataSet-3	DataSet-4	DataSet-5	DataSet-6
K-means	1	0.255	2.563	0.696	0.900	0.918	1.362
	2	93.387%	49.175%	76.766%	38.765%	57.825%	57.243%
	3	N/A	0%	0%	0%	0%	0%
DBSCAN	1	123.631	152.900	150.085	151.294	150.503	150.556
	2	100.000%	100.000%	99.965%	99.995%	99.999%	99.976%
	3	N/A	95.020%	96.960%	95.627%	95.407%	96.147%
IDBSCAN	1	21.136	30.016	39.966	35.905	37.636	37.897
	2	99.997%	100.000%	99.927%	99.984%	99.938%	99.938%
	3	N/A	95.413%	97.787%	96.133%	96.400%	96.647%
KIDBSCAN	1	20.903	29.717	39.558	35.919	37.347	37.627
	2	99.982%	99.989%	99.909%	99.982%	99.949%	99.924%
	3	N/A	95.340%	97.725%	96.132%	96.403%	96.633%
NPUST	1	9.432	8.720	12.516	12.499	13.199	14.018
	2	99.956%	99.960%	99.897%	99.966%	99.720%	99.908%
	3	N/A	95.373%	97.707%	96.215%	96.491%	96.695%

Table 2. Comparisons of number of queries (in millions) of NPUST, DBSCAN, IDBSCAN and KIDBSCAN using 115,000 datasets with DataSets 2-6; the DataSet 1 has 100,000 points

Algorithm	DataSet-1	DataSet-2	DataSet-3	DataSet-4	DataSet-5	DataSet-6
DBSCAN	10000.000	12194.255	12079.140	12136.755	12104.555	12086.040
IDBSCAN	1710.800	2394.070	3212.525	2872.585	3016.680	3039.680
KIDBSCAN	1693.810	2399.751	3166.203	2889.996	3001.489	3022.407
NPUST	600.708	328.697	724.184	727.906	766.822	774.928

are listed. Table 1 presents the clustering experimental results from NPUST, K-means, DBSCAN, IDBSCAN and KIDBSCAN by using 100,000 (DataSet 1) and 115,000 (DataSets 2-6) points datasets, and indicates that the proposed NPUST algorithm had a lower run-time cost than DBSCAN, IDBSCAN and KIDBSCAN. Additionally, NPUST also yielded almost the same high quality of data clustering as DBSCAN, IDBSCAN and KIDBSCAN. Fig. 4 depicts the datasets obtained by NPUST. Experimental results demonstrate that NPUST can accurately recognize clusters for arbitrary shapes. Although K-means cannot identify arbitrary shapes, or filter out noise, it has the lowest run-time cost.

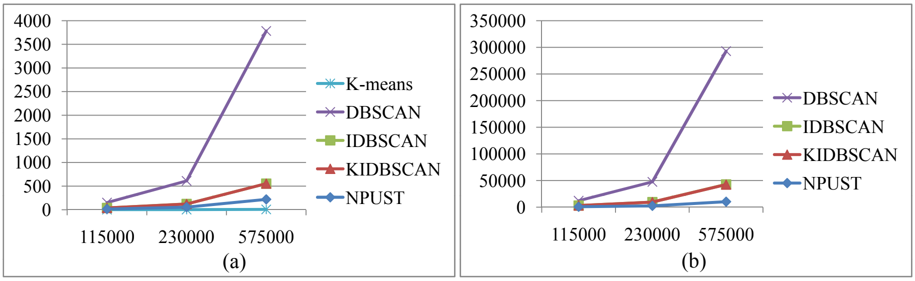


Fig. 5. (a) Performance comparison of K-means, DBSCAN, IDBSCAN, KIDBSCAN and NPUST using DataSet 6 with various data sizes; x-axis indicates data sizes; y-axis represents the run-time cost (in seconds) (b) The number of queries comparison of DBSCAN, IDBSCAN, KIDBSCAN and NPUST using DataSet 6 with various data sizes; x-axis denotes data sizes; y-axis represents the number of queries (in millions)

Table 2 lists the number of queries run by NPUST, DBSCAN, IDBSCAN and KIDBSCAN with 100,000-point (DataSet 1) and 115,000-point (DataSets 2-6) datasets, revealing that the proposed NPUST has the lowest number of queries due to its use of the partition space technique. The number of queries in DBSCAN is almost always n^2. IDBSCAN and KIDBSCAN sample the points as the *seed* to decrease the huge number of queries, while KIDBSCAN start cluster from high-density points. Fig. 5a shows the run-time comparison of NPUST, K-means, DBSCAN, IDBSCAN and KIDBSCAN with DataSet 6 with various data sizes. These experimental results indicate that increasing data size raises the difference in run time between K-means, DBSCAN, IDBSCAN, KIDBSCAN and NPUST. Fig. 5b displays a comparison of the number of queries run by DBSCAN, IDBSCAN, KIDBSCAN and NPUST using DataSet 6 with various data sizes. An increasing data size leads to increasing difference in the number of queries run by DBSCAN, IDBSCAN, KIDBSCAN and NPUST, as revealed in Fig. 5b.

6 Conclusion

This work presents an improved KIDBSCAN clustering algorithm for handling large databases. The proposed NPUST algorithm extends the existing KIDBSCAN by incorporating an improved partitioning technique to reduce time taken by neighborhood search. Experimental results have established the partitioning method significantly shortens the run-time of clustering. NPUST outperforms DBSCAN, IDBSCAN and KIDBSCAN in terms of execution time, without losing clustering quality. Although K-means performs clustering very quickly, it is not practical because it cannot filter out noise. The proposed NPUST clustering algorithm is more efficient and effective than previous algorithms.

Acknowledgments. The author would like to thank the National Science Council of Republic of China for financially supporting this research under contract no. NSC 96-2221-E-020-027.

References

1. Wang, T.P., Tsai, C.F.: GDH: An Effective and Efficient Approach to Detect Arbitrary Patterns in Clusters with Noises in Very Large Databases. In: Degree of master at National Pingtung University of Science and Technology, Taiwan (2006)
2. Tsai, C.-F., Liu, C.-W.: KIDBSCAN: A New Efficient Data Clustering Algorithm for Data Mining in Large Databases. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 702–711. Springer, Heidelberg (2006)
3. Tsai, C.-F., Yen, C.-C.: ANGEL: A New Effective and Efficient Hybrid Clustering Technique for Large Databases. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 817–824. Springer, Heidelberg (2007)
4. McQueen, J.B.: Some Methods of Classification and Analysis of Multivariate Observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231. AAAI Press, Menlo Park (1996)
6. Borah, B., Bhattacharyya, D.K.: An Improved Sampling-Based DBSCAN for Large Spatial Databases. In: Proceedings of International Conference on Intelligent Sensing and Information Processing, Chennai, India, pp. 92–96 (2004)
7. Zhang, T., Ramakrishnan, R.: BIRCH: An efficient Data Clustering Method for Very Large Databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 103–114. ACM Press, Montreal (1996)
8. Guha, S., Rastogi, R., Shim, K.: CURE: An Efficient Clustering Algorithm for Large Databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 73–84. ACM Press, Seattle (1998)
9. Guha, S., Rastogi, R., Shim, K.: ROCK: A Robust Clustering Algorithm for Categorical Attributes. In: Proceedings of the 15th International Conference on Data Engineering, pp. 512–521. IEEE Press, Sydney (1999)
10. Karypis, G., Han, E.H., Kumar, V.: CHAMELEON: A Hierarchical Clustering Using Dynamic Modeling. IEEE Computer: Special Issue on Data Analysis and Mining 32(8), 68–75 (1999)
11. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 94–105. ACM Press, Seattle (1998)

A Method for Automatic Discovery of Reference Data

Lukasz Ciszak

Institute of Computer Science, Warsaw University of Technology,
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
L.Ciszak@ii.pw.edu.pl

Abstract. The data quality assessment process consists of several phases; the first phase is the data profiling step. The result of this step is the set of the most current metadata describing the examined data set. We present here a method for automatic discovery of reference data for textual attributes. Our method combines the textual similarity approach with the characteristics of attribute value distribution. The method can discover the correct reference data values also in situations where there is a large number of data impurities. The results of the experiments performed on real address data prove that the method can effectively discover the current reference data.

1 Introduction

Data profiling is the first step of the data quality assessment process. There is a class of profiling tasks that concentrate on the analysis of the data set in order to discover a set of values which are assumed to be correct in terms of a given attribute. This set of data is called reference data and may be perceived as a domain constraint for a given attribute.

We concentrate here on the process of automatic discovery of the most up to date reference data. We propose a novel automatic method that combines the techniques of textual similarity with the characteristics of a value distribution within the analyzed data set. The discovered reference data may be merged with the existing data set in order to obtain their most recent version.

The structure of this paper is as follows: the first chapter presents briefly the process of data quality assessment and related research in the field of reference data discovery. The second chapter presents the method of automatic discovery of the reference data for address related attributed in a personal data database. The third chapter presents the results of the experiments on real and artificial data sets performed using the method and a discussion of algorithm effectiveness.

1.1 Data Quality Assessment Overview

Data quality is defined as 'fitness for use' [8]. It defines the extent to which the data can satisfy the business processes they were collected for. The quality may

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 797–805, 2009.

be split into several other dimensions such as accuracy, completeness, correctness, relevancy, timeliness, or consistency [10]. Data quality assessment is the process whose aim is to determine the quality of data. The first step of this process is data analysis, also known as profiling. The goal of profiling is to gather the most current information about the examined data. The collected and the existing metadata are then merged in order to create the most accurate metadata set. The new metadata may then be used in data cleaning applications. Reference data is a set of values which are considered valid for a given attribute and may be used in the definition of domain constraints for this attribute. In some situations, the existing metadata may be obsolete, or unavailable. In such cases, the profiling process should result in the discovery of the reference data.

1.2 Related Work

Current approaches to the discovery of reference data involve manual inspection of the results of the profiling process, namely results of the 'Distribution', 'Top N', and 'Bottom N' tasks [7,6,8]. However, as misspellings, mistypings, or other data impurities may occur within the data set, this method may wrongly identify incorrect versions of the reference data elements as correct and belonging to the reference data set. To our knowledge no research results devoted specifically to the automatic discovery of reference data have been published.

The area of research related to this problem is the area of duplicate detection, also known as the 'merge/purge' problem. The methods in this area focus on the discovery of attribute values that refer to the same real world object. Hernandez [5] proposes a method based on a textual similarity of elements. In this method, the elements are sorted alphabetically which causes the misspelled/mistyped versions of reference data elements appear next to the correct elements. Thanks to this fact, the method can effectively match duplicates. Winkler [11] utilizes the statistical record linkage method based on the Fellegi-Sunter theorem. Records are matched basing on their textual (or other) similarity and the probability of duplicate existence. The method takes also in account the rules defined by a data analyst. Bilenko and Mooney [3] propose a method that utilizes machine learning techniques. In their method, the algorithm learns the correct textual similarity distance threshold that is later used to match duplicate items. Authors in [1] propose a record linkage method aiming at limiting the number of false positives, that is, the records incorrectly identified as duplicates. Their method takes into account the hierarchies that the examined data elements may create.

In contrast, our approach automatically discovers the reference data with little human intervention, which can speed up the data profiling phase. It also takes into account the impurities and errors in data. Therefore, it can eliminate incorrect values with large number of occurrences that may appear in the examined data set.

2 The Proposed Approach

In this section, we introduce our approach to the automatic discovery of the reference data values. The approach combines the features of textual similarity approach with the statistical approach. The method takes into account the characteristics of value distribution within the data set. The output is the set of reference data which can be then confronted with the existing metadata.

2.1 Method Description

Let $S = \{s_1, s_2, \ldots, s_n\}$ denote the set of values of a given attribute. Each element s_n of the set S is described by a pair of values (v_n, o_n), where v_n is the text string representing the value and o_n is the number of occurrences of a given value within the data set. Let R denote the reference data set, I the set of incorrect versions of the R set elements, and N the set of noise values.

Definition 1. *The function $M : S \times S \rightarrow \mathbb{R}^2$ is a distance metric between the elements of the set S, such that*

$$M(s_m, s_n) = (d_{mn}, r_{mn})$$

where:

$$d_{mn} = D_w(v_m, v_n) \ , \ d_{mn} \in [0, 1] \ ,$$
$$r_{mn} = max(\frac{o_m}{o_n}, \frac{o_n}{o_m}) \ , \ r_{mn} \in [1, \infty)$$

D_w is the Jaro-Winkler measure of similarity between two text strings [11]. It takes into account the number of letter transpositions and deletions needed to transform one text string into the other. It varies from 0 to 1 and is equal 0 for two completely different strings, whereas for two identical strings it is equal 1. For instance, $D_w('Warsaw','Warswa') = 0.97$.

Definition 2. *Similarity relation '$\backsim$' over the set S is a relation defined in that way that:*
$s_m \backsim s_n \Leftrightarrow d_{mn} \geq \varepsilon \wedge r_{mn} \geq \alpha \wedge o_m < o_n$
where ε is a similarity threshold parameter and α is an occurrence ratio threshold parameter.

We assume that the set S has following features:

$$S = R \cup I \cup N \tag{1}$$

$$\forall i \in I \exists p \in R \colon i \backsim p \tag{2}$$

$$\nexists n \in N, p \in R \colon n \backsim p \tag{3}$$

The '$\backsim$' relation over the set S may be represented as a directed weighted graph [9]. The elements of the set are represented as the nodes of the graph, whereas the arcs of the graph represent the similarity relation. The weight of the arc between elements s_m and s_n is equal $M(s_m, s_n)$.

Definition 3. *The indegree of a node s, denoted by* $deg^{-}(s)$*, is the number of incoming arcs.*

Definition 4. *The outdegree of a node s, denoted by* $deg^{+}(s)$*, is the number of outgoing arcs.*

Definition 5. *A sink is a node for which* $deg^{+}(s) = 0 \wedge deg^{-}(s) > 0$*;*

Definition 6. *An isolated node is a node for which* $deg^{+}(s) = 0 \wedge deg^{-}(s) = 0$

We have observed that all examined address datasets displayed the features of scale-free networks [2] as there were hub nodes – sink nodes with significantly high indegree.

We assume that the reference dataset consists of the values stored in hub nodes and isolated nodes with a large number of occurrences. Remaining nodes represent incorrect (misspelled, mistyped, etc.) versions of values from the reference data set. Isolated nodes with low number of occurrences of value are assumed to be noise.

2.2 Algorithm Definition

The algorithm discovers the reference data set by analyzing the structure of the graph. The parameters of the algorithm are:

1. ε: textual similarity threshold, that is, the minimum textual similarity between two strings allowing to treat them as similar,
2. α: occurrence threshold, that is, the minimum ratio of occurrences of two attributes allowing to determine the direction of the similarity relation.

The algorithm consists of the following steps:

1. Distance matrix calculation: in this steps we calculate the similarity matrix $A = (a_{ij})$ where $a_{ij} = M(s_i, s_j)$ using the M metric defined in previous section.
2. Graph construction: using the similarity matrix we construct the graph satisfying algorithm parameters ε and α.
3. Graph reduction: for each node we remove all the arcs except the one having the maximum weight, that is, the textual similarity between the nodes it connects is maximal and they have high occurrence ratio. Also, in this step we remove the out arcs for nodes having the indegree greater or equal than the defined limit for sink nodes.
4. Graph merge: to each sink node we connect its indirect neighbors, that is, all the nodes from which the given sink node is reachable.
5. Reference set extraction: the reference data set consists of values in the sink nodes and values having a large number of occurrences, assigned to isolated nods.

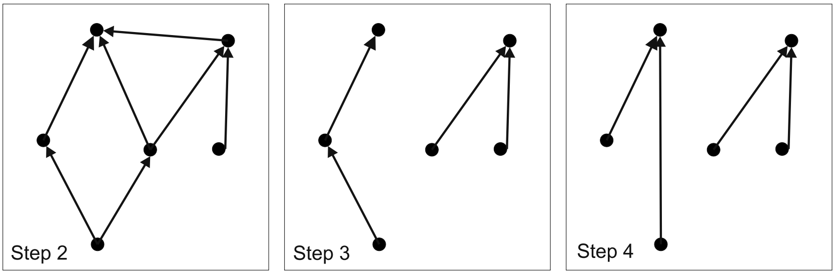


Fig. 1. The fragment of the graph structure after the application of steps 2., 3., and 4. of the algorithm

3 Experimental Results

We have performed a series of experiments in order to determine how well the method can discover reference data containing all the 'real' reference data set values. We have defined following algorithm effectiveness measures:

1. p_m the percentage of the reference values that were not discovered by the algorithm.
2. p_r the percentage of redundant values, that is, values that were identified as the reference data, but were in fact incorrect versions of the 'real' reference data set elements.

The measures are defined as follows:

$$p_m = 100 - \frac{n_c}{n_r} \times 100 \tag{4}$$

$$p_r = \frac{n_a - n_c}{n_a} \times 100 \tag{5}$$

where n_c is the number of discovered true reference values, n_r is the number of the number of true reference values, and n_a is the number all discovered reference values.

The result of each algorithm run for a pair input parameters were the values of the aforementioned measures.

The tests were performed on two data sets. The first data set was an artificial set which consisted of the names of the fifty largest American cities combined with a set of incorrect values. The incorrect values were generated by damaging the correct names in accordance with the error typology statistics published by Giebułtowicz [4]. The percentage of incorrect values reflected that typical for the real world data from and was equal 10%. The numbers of the correct value occurrences were generated according to the Zipf law. In this case, the correct city name were the reference data. Table 1 contains an excerpt from this data set.

Table 1. Example data from the artificial data set

Value	Number of occurrences	Reference value
New York City	5000	New York City
New uork City	3	New York City
New YorkXCity	1	New York City
NewYork City	7	New York City
New York Citty	10	New York City

Table 2. Example data from the real data set

Value	Number of occurrences	Reference value
WARSZAWA	6160	WARSZAWA
WARSAWA	1	WARSZAWA
WARSYAWA	1	WARSZAWA
WARSZA	1	WARSZAWA
WARSZAW	4	WARSZAWA
WARSZWA	8	WARSZAWA

Table 3. The dependency between the algorithm parameters and the p_r measure

p_r	α									
ε	1	2	3	4	5	6	7	8	9	10
0.0	0	0	3	3	1	1	1	1	2	2
0.1	0	0	3	3	1	1	1	1	2	2
0.2	0	0	3	3	1	1	1	1	2	2
0.3	0	0	3	3	1	1	1	1	2	2
0.4	0	0	3	3	1	1	1	1	2	2
0.5	0	0	3	2	1	1	1	1	2	2
0.6	0	0	2	2	1	1	1	1	2	2
0.7	1	1	4	5	5	7	7	8	8	8
0.8	2	2	3	4	5	6	6	6	6	7
0.9	4	4	5	6	7	7	7	8	8	8
1.0	15	15	15	15	15	15	15	15	15	15

The second data set is a real data set of Polish cities taken from an Internet survey. We have inspected the set manually and discovered the reference data set. In addition, all the values identified as incorrect were assigned a correct value from the identified reference data set. Table 2 contains an excerpt of the data set.

In terms of the algorithm effectiveness we have obtained qualitatively the same results both from the artificial and the real world data set which confirms the assumptions about the data. The algorithm has proven to work correctly; it managed to identify almost all the reference data values keeping at the same time the redundant values percentage low.

Table 4. The dependency between the algorithm parameters and the p_m measure

p_m					α					
ε	1	2	3	4	5	6	7	8	9	10
0.0	98	98	97	96	95	95	95	95	95	94
0.1	98	98	97	96	95	95	95	95	95	94
0.2	98	98	97	96	95	95	95	95	95	94
0.3	98	98	97	96	95	95	95	95	95	94
0.4	98	98	97	96	95	95	95	95	95	94
0.5	98	98	96	96	95	95	95	95	94	94
0.6	97	97	95	94	94	93	92	92	91	91
0.7	93	90	86	83	82	80	78	77	75	74
0.8	62	57	48	43	40	37	33	31	30	29
0.9	9	8	7	6	6	6	5	5	4	4
1.0	0	0	0	0	0	0	0	0	0	0

For the artificial data, the algorithm discovered 100% values of the original reference data set, while the percentage of redundant values was equal to 4%. For the real world data, the algorithm discovered 96% of the reference data set values, while the percentage of redundant values was as low as 8%. These values were obtained for the algorithm parameters $\varepsilon = 0.9$ and $\alpha = 9$. The results for the real-world data set are contained in Table 3 and Table 4.

Figure 2 shows the dependency between the measures p_m and p_r, and the parameter ε for $\alpha = 9$. It shows that with the growth of the ε parameter, the percentage of undiscovered reference value p_m is decreasing. However, the larger the ε parameter is, the more redundant values are discovered as reference data values. Therefore, the larger the ε parameter is, the more values are assumed to be unique and, hence, belong to the reference data set.

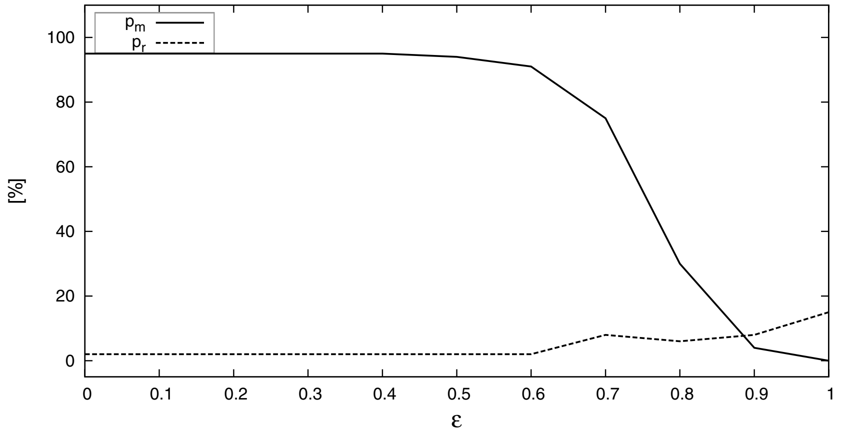


Fig. 2. The dependency between the algorithm parameter ε and the measures p_m and p_r

The algorithm displayed slightly better results for the artificial data set. This was caused by the fact that it contained only fifty reference values, and the textual similarity measures between each pair of them was low. In contrast, the real data set contained a large number of unique reference values, and some of them were textually similar to one another.

The major drawback of this method is that in case of reference values that have relatively low number of occurrences within the data set and which are similar to other values of the set, may be identified as incorrect versions of the reference data set values and eliminated. Therefore, method improvement should involve examining the context of the cleaned value.

4 Conclusion

In this paper, we have presented a novel approach to the discovery of reference data set that combines two approaches - textual similarity and value distribution. Contrary to other methods and approaches in this area, our method is automatic and needs very little human intervention. Experimental results show that the method can effectively discover reference data set from the examined data, even if there is large percentage of data impurities. The method gives best results when the examined data contain a low number reference elements that are distant in terms of the textual similarity.

We plan to modify our method so that also the context of the attribute is taken into account by using functional dependencies. This could effectively solve the problem with the infrequently occurring reference values.

References

1. Ananthakrishna, R., Chaudhuri, S., Ganti, V.: Eliminating Fuzzy Duplicates in Data Warehouses. In: Proceedings of the 28th international conference on Very Large Data Bases, pp. 586–597 (2002)
2. Barabási, L., Bonabeau, E.: Scale-Free Networks. Scientific American 288, 60–69 (2003)
3. Bilenko, M., Mooney, R.: Adaptive Duplicate Detection Using Learnable String Similarity Measures. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2003)
4. Giebułtowicz, M.: Polish spelling errors categorization. In: Proceedings of the International Interdisciplinary Technical Conference of Young Scientists (2008)
5. Hernández, M., Stolfo, S.: Real-world data is dirty: Data cleansing and the merge/purge problem. Data Mining and Knowledge Discovery 2, 9–37 (1998)
6. Kimball, R., Caserta, J.: The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data, p. 525. Wiley John & Sons, Chichester, ISBN: 9780764567575
7. Lindsey, E.: Three-Dimensional Analysis. Data Profiling Techniques. Data Profiling LLC, p. 242 ISBN: 9780980083309

8. Maydanchik, A.: Data Quality Assessment. Technics Publications, p. 336 ISBN: 9780977140022
9. Monge, A.: Adaptive detection of approximately duplicate records and the database integration approach to information discovery. PhD thesis, University of California (1997)
10. Olson, J.: Data Quality: The Accuracy Dimension, p. 300. Morgan Kaufmann, San Francisco, ISBN: 9781558608917
11. Winkler, W.E.: The state of record linkage and current research problems. Statistics of Income Division, Internal Revenue Service Publication R99/04 (1999)

Mining Non-redundant Reclassification Rules

Li-Shiang Tsay[1] and Seunghyun Im[2]

[1] North Carolina A&T State Univ., School of Technology,
203 Price Hall, Greensboro, USA
[2] Univ.of Pittsburgh at Johnstown, Dept. of Comp. Science,
348 Budfield St, Johnstown, USA
ltsay@ncat.edu, lishiangtsay@yahoo.com, sim@pitt.edu

Abstract. The increased competition faced by today's companies can wield data mining tools to extract actionable knowledge and then use it as a weapon to outmaneuver competitors and boost revenue. Mining reclassification rules is a way to model actionable patterns directly from a given data set. The previous work on reclassification rule mining has shown that they are effective when variables are weakly correlated. However, when the data set is correlated, some redundant rules are in the result set. This problem becomes critical for discovering rules in correlated data which may have long frequent factor-sets. In this paper, we investigate properties of reclassification rules and offer a new method to discovery a set of non-redundant reclassification rules without information loss.

1 Introduction

Discovering actionable patterns in a data set is an important data-mining problem. To address this issue, a vast number of knowledge discovery techniques have been developed which extracted information in the form of association rules, classification rules, clusters, and many other representations. These various techniques can learn rules that summarize the data. In general, these types of rules are not directly useful for the business sector. To find truly interesting and actionable patterns, human involvement in the pattern evaluation and interpretation is vital. For example, a marketing analyst may want to analyze if and how the cross-selling effect differs across several bundle packages in a database of customer transactions. Such information can lead to increased sales by helping retailers do careful marketing. Association rule mining may be used to identify related products from existing products, but not the cross-selling effect. Can this problem be solved by another standard learning method, such as classification algorithm to model actions and effects? Unfortunately, the standard methods treat populations separately and tend to fragment the knowledge space. Each group is seldom interesting by itself without meaningful context comparisons with others. The classical mining paradigm itself poses a major obstacle for this actionability analysis task.

A key and non-trivial issue in actionable pattern mining is the ability to acquire all facts, understanding their causes and effects, and listing all potential solutions and the responding effects. Several papers proposed to model the actionability patterns in either a domain-dependent way [8] or post-process analyzing rules manner [1] [4-5]

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 806–815, 2009.

[9-12] [15-17]. A Domain dependent approach may work well for the specific application, but it can not be generalized to other application domains. A post-process analyzing approach generally utilizes a standard data mining algorithm first and then on the basis of these mined results the actionable plans are determined. It may work well for various applications, but it is no guarantee that the extracted patterns in the first step will lead to actionability.

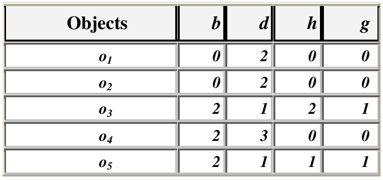
Some recent research [2-3] [13] takes a different approach, object-based analyzing, to generate actionable patterns directly from a data set without using pre-existing rules. Such approach may work well for a dataset with predetermined discovery's target, but it may insufficient to know the space of all actionable patterns. Later on, *Reclassification Rules* [14], the action-effect model, provides a way to identify the space of all actionable patterns existing in the database that satisfy some predefined thresholds. The method proposed in [14] enumerates all frequent factor-sets and then using these for rule generation produces some redundant rules, as we will show later. To address this issue, we focus on generating a fairly small set of all non-redundant reclassification rules without losing the information captured in the original rules.

To address above concerns, there are two possible approaches: (1) compressing the frequent factor-sets into a relatively small set, i.e., identifying a basic set of factor-sets from which all other frequent factor-sets can be derived, and (2) reducing the number of rules, i.e., identifying a set of optimal rules having minimal antecedent and maximal consequent to pruning redundant rules. In this paper, we developed a similar approach to the concept of frequent closed item-sets [6] in order to limit the search space for a concise set of frequent factor-sets. Such set of frequent patterns suffices to determine a reduced set of reclassification rules without information loss. We proposed a new algorithm, called *StrategyGeneratorIII*, to find such set of frequent closed factor-sets and then form a set of non-redundant rules.

2 Reclassification Model

An information system [7] is used for representing knowledge and is defined as a pair $S = (U, A)$, where U is a nonempty, finite set of objects, and A is a nonempty, finite set of attributes, i.e. $a : U \to V_a$ is a function for any $a \in A$, where V_a is called the domain of a. Elements of U are called objects. In this paper, for the purpose of clarity, objects are interpreted as customers. Attributes are interpreted as features such as offers made by a company. Table 1 shows an example of an information system, which consists of 5 objects $\{o_1, o_2, o_3, o_4, o_5\}$ described by 4 attributes $\{b, d, h, g\}$.

Table 1. Information system



Objects	b	d	h	g
o_1	*0*	2	*0*	*0*
o_2	*0*	2	*0*	*0*
o_3	2	*1*	2	*1*
o_4	2	3	*0*	*0*
o_5	2	*1*	*1*	*1*

Factor-Set. Any set of factors is called a factor-set F. It is partitioned into constant factors F_S and flexible factors F_F. Let us assume that (v_1, $v_2 \in V_a$). Then, the expressions (a, $v_1 \rightarrow v_2$) are elements of $F_S \cup F_F$. If $v_1 \neq v_2$, then (a, $v_1 \rightarrow v_2$) refers to a flexible factor and it denotes the fact that the value of attribute a can be changed from v_1 to v_2 for a number of objects in S. Similarly, the term (a, $v_1 \rightarrow v_2$)(o) means that the attribute value $a(o) = v_1$ can be changed to $a(o) = v_2$ for the object o. If $v_1 = v_2$, (a, $v_1 \rightarrow v_2$) refers to a constant factor and its value does not change. A factor-set is defined as any conjunction of factors where each factor is associated with a different attribute, for example, F=(a_1, $v_1 \rightarrow v_2$), (a_2, $v_1 \rightarrow v_2$), …, (a_n, $v_1 \rightarrow v_2$), where $(\forall i \leq n)(\forall j \leq n)[(i \neq j) \rightarrow (a_i \neq a_j)]$.

Frequent Factor-Set. A factor-set ζ which supports the user specified minimum *left support* threshold $s_1 \geq 0$ and minimum *right support* threshold $s_2 \geq 0$ is said to be *frequent.* The collection of all frequent factor-sets supporting these two thresholds is denoted by $\mathscr{F}(s_1, s_2)=\{\zeta_1, \zeta_2, \ldots, \zeta_n\}$. $supL(\zeta)$ denotes left support and $supR(\zeta)$ denotes right support of ζ. A frequent factor-set ζ in S is defined as a term

$$\zeta= [(a_1, \alpha_1 \rightarrow \beta_1) \wedge (a_2, \alpha_2 \rightarrow \beta_2) \wedge \ldots \wedge (a_p, \alpha_p \rightarrow \beta_p)], \text{ where } (\forall i \leq p)(\forall j \leq p)[(i \neq j) \rightarrow (a_i \neq a_j)],\ supL(\zeta) \geq s_1, \text{ and } supR(\zeta) \geq s_2 .$$

The *Left Support* defines the domain of a factor-set which identifies objects in U on which the factor-set can be applied. The larger its value, the more interesting the factor-set will be for a user. The left hand side of the factor-set ζ, is defined as $P_L = \{\alpha_1, \alpha_2, \ldots, \alpha_p\}$. The domain $Dom_S(P_L)$ is a set of objects in S that exactly match P_L. The left support $supL$ of the factor-set ζ is defined as $supL(\zeta) = Card[Dom_S(P_L)] /Card[U]$.

The *Right Support* shows how well the factor-set is supported by objects in S. The higher its value, the stronger case of the reclassification effect will be. The right hand side of the factor-set ζ is defined as $P_R = \{\beta_1, \beta_2, \ldots, \beta_p\}$. The domain $Dom_S(P_R)$ is a set of objects in S that exactly match P_R. The *right* support $supR$ of the factor-set ζ is defined as $supR(\zeta) = Card[Dom_S(P_R)]/Card[U]$.

Reclassification Rule. The set of all candidate rules is constructed from frequent factor-sets. The number of factors in a factor-set will be called the *length* of the factor-set. A factor-set of the length k is referred to as a *k-factor-set.* Each rule is extracted through *k-factor-set* having at least two flexible factors f_F, when $k \geq 2$. A candidate rule which confidence is greater than the user specified minimum confidence c is called a strong rule and it belongs to $\mathscr{R}(s_1, s_2, c)$.

A reclassification rule $\mathscr{r}$ in S is defined as a statement

$$\mathscr{r} = [[(a_1, \alpha_1 \rightarrow \beta_1) \wedge (a_2, \alpha_2 \rightarrow \beta_2) \wedge \ldots \wedge (a_p, \alpha_p \rightarrow \beta_p)] \Rightarrow [(b_1, \phi_1 \rightarrow \psi_1) \wedge (b_2, \phi_2 \rightarrow \psi_2) \wedge \ldots \wedge (b_q, \phi_q \rightarrow \psi_q)]$$

where, for $i= 1, 2, \ldots p$ *and* $j=1,2, \ldots q$, *all* a_i, $b_j \in A$ *are different.* The term $(a_i, \alpha_i \rightarrow \beta_i)$ is called the action premise (antecedent) and $(b_j, \phi_j \rightarrow \psi_j)$ is called the effect (consequent) of the reclassification rule. We say that objects $o_1, o_2 \in U$ support the reclassification rule r in S, if and only if:

- $(\forall i \le p)\ [[a_i(o_1) = \alpha_i] \wedge [a_i(o_2) = \beta_i]]$,
- $(\forall \mathrm{j} \le q)\ [[b_j(o_1) = \phi_1] \wedge [b_j(o_2) = \psi_i]]$.

Statistical significance of a rule r is expressed in terms of *left support* and *right support*, denoted as $supL(r)$ and $supR(r)$, respectively. The *left support* measure defines the range of the rule, i.e., the proportion of objects which are applicable, while the *right support* measure defines the feasibility of the rule, i.e., the proportion of objects which are role models for others. Because supports indicate the frequency of the factor-set occurring in the rule, then $supL(r)$ and $supR(r)$ are basically the same as the supports for the factor-set ζ, i.e., $supL(r)=supL(\zeta)$ *and* $supR(r)=supR(\zeta)$.

The strength of a reclassification rule r is characterized by confidence, which is denoted by conf(r). Statistical significance of the *antecedent* of a rule r is called *antecedent-right-support* and *antecedent-left-support* which are denoted as $a\text{-}supL(r)$ and $a\text{-}supR(r)$, respectively. $a\text{-}supL(r)$ is defined as the number of objects in S which have property α_i, for $i = 1, 2, \ldots, p$. $a\text{-}supR(r)$ is defined as the number of objects in S that have property β_j, for $j = 1, 2, \ldots, q$. The *confidence* of a reclassification rule r in S is defined as $conf(r) = [supL(\zeta) \times supR(\zeta)] \div [a\text{-}supL(r) \times a\text{-}supR(r)]$.

3 StrategyGeneratorII

Algorithm *StrategyGeneratorII* mines a set of all frequent factor-sets and then constructs reclassification rules by considering changes of values within attributes of atomic expressions from which more complex expressions are built. It consists of two main steps: (1) generate all frequent factor-sets, (2) generate strong reclassification rules.

Generate all Frequent Factor-Sets. This step computes the frequent factor-sets in the data set through several iterations. In general, iteration k computes all factor-sets of length k from factor-set of length k-1. It uses a breadth-first bottom-up approach where frequent subsets are extended one factor at a time. In each iteration, we generate new candidates from frequent factor-sets found in the previous iteration; the left support and right support for each candidate is then computed and tested against the user-specified thresholds. The total number of iterations needed by the algorithm is $k_{max}+1$, where k_{max} is the maximum size of frequent factor-sets.

Referring back to Table 1, all frequent factor-sets with minimum $supL$ = 20% and $supR$ = 40% are shown in Table 2. Stable atomic factors can not be solely used either in the rule consequent or in the rule premise for rule candidates' generation. If a factor-set has less than two f_{F} , the factor-set will be pruned from the frequent list. In our example, we have seventeen disqualified factor-sets such as [(b,2→2) (g,0→1)], [(h,0→0)(b,2→0)], etc., those factor-sets are marked with an asterisk on the right corner in the Table 2.

Table 2. Generating all Frequent Factor-Sets

supL	***supR***	**Frequent Factor-sets**
0.20	**0.40**	[(b,2→2)(g0→1)]*, [(h,0→0)(b,2→0)]*, [(g,0→0)(b,2→0)]*
0.40	**0.40**	(b, 0→0)*, (d, 1→1)*, (d, 2→2)*, (g, 1→1)*, (d, 1→2)*, (d, 2→1)*, [(b,0→0)(d,2→2)]*, [(b,0→2) (d,2→1)], [(b,0→2)(g,0→1)], [(b,2→0)(d,1→2)], [(b,2→0)(g,1→0)], [(d,1→2)(g,1→0)], [(d,2→1) (g,0→1)], [(b,0→2)(d,2→1)(g,0→1)], [(b,2→0)(d,1→2)(g,1→0)]
0.40	**0.60**	(b, 0→2)*, (g, 1→0)*
0.60	**0.40**	(b, 2→0)*, (g, 0→1)*
0.60	**0.60**	(b, 2→2)*, (h, 0→0)*, (g, 0→0)*

Generate Strong Rules. A reclassification rule is extracted by partitioning the factor-set Z into two non-empty subsets X and Y, where ($X, Y \subseteq Z$), ($X, Y \not\subset F_S$), and $X=Z-Y$, such that $X \Rightarrow Y$. A level-wise approach is utilized for generating rules, where each level corresponds to the number of factors belonging to the rule *premise*. The rule-premise is extended one factor at a time. It starts with one factor on the rule-premise, such as X. If the rule confidence is above the predetermined threshold, it is called a strong rule and marked with a positive mark. The principle of *minimum description length* is also adopted to identify the general rules. Therefore, in the next iteration, it generates new candidates from only unmarked rules found in the previous iteration by moving one of the consequence-type factors to the rule premise, now X is a 2-factor-set. Repeat the previous step until there is only one flexible factor on the rule consequent, such as Y. By doing this, the result set of rules are those with the shortest length of premise, not all the lengths.

Referring back to Table 1, the set of all strong reclassification rules with confidence above or equal to minimum *conf* = 90% are shown in Fig. 1. From factor-set [(b, 2→0), (d, 1→2), (g, 1→0)], one strong rule r_2 is generated in the first loop, so it will be not considered in the second loop. Rules r_1 and r_3 are not strong rules, and then candidate rule r_4 is generated by merging the premises of both rules. This process is shown in the left hand side of the Fig. 1.

Rules Generation

Frequent Factor-Set: [(b,2→0)(d,1→2)(g,1→0)]

Loop 1: // for illustration, each rule is labeled

r_1:(b,0→2)⇒(d,2→1)(g,0→1) $(2\times2)/(2\times3) < 0.9$

r_2:(d,2→1)⇒(b,0→2)(g,0→1) $(2\times2)/(2\times2) \geq 0.9$ "+"

r_3:(g,0→1)⇒(d,2→1)(b,0→2) $(2\times2)/(3\times2) \geq 0.9$

Loop 2:

r_4:(b,0→2)(g,0→1)⇒(d,2→1) $(2\times2)/(2\times2) \geq 0.90$ "+"

All Strong Reclassification Rules

Rules	**supL**	**supR**	**conf**
R_1: (d, 2→1) ⇒ (b, 0→2)	2/5	2/5	100%
R_2: (d, 1→2) ⇒ (b, 2→0)	2/5	2/5	100%
R_3: (d, 1→2) ⇒ (g, 1→0)	2/5	2/5	100%
R_4: (d, 2→1) ⇒ (g, 0→1)	2/5	2/5	100%
R_5: (d, 2→1) ⇒ (b, 2→0) (g, 0→1)	2/5	2/5	100%
R_6: (b, 0→2) (g, 0→1) ⇒ (d, 2→1)	2/5	2/5	100%
R_7: (d, 1→2) ⇒ (b, 2→0) (g, 1→0)	2/5	2/5	100%
R_8: (b, 2→0) (g, 1→0) ⇒ (d, 1→2)	2/5	2/5	100%

Fig. 1. Rule Generation and all strong reclassification rules

Limitations. Algorithm *StrategyGeneratorII* is efficient for mining reclassification rules from weakly correlated data. However, when the data set is correlated such as in our example, some redundant rules, i.e., rules bringing the same information, are in the resulting set. In Fig. 1, rules R_2 and R_3 do not add any information to rule R_7 since all these rules have identical antecedent, left support, right support, and confidence. These rules are redundant compared to rule R_7, which is the most interesting rule from the analyst's point of view because it summarizes these three rules. The rule R_7 has the longest consequent among the three rules. A major challenge is that such algorithm often generates a large number of frequent factor-sets satisfying the minimum support thresholds, especially when minimum left support and right support are set low. As mentioned in the introduction section, the more amounts of frequent factor-sets, the more rules there will be. The longer the length of the longest frequent factor-set, the more number of redundant rules will be presented to the users. This issue has become more and more counteractive when the rules are extracted from correlated data which may have long frequent factor-sets.

To tackle the above issue, two possible approaches for generating a set of non-redundant rules without losing the information captured in the original rules are: (1) *identifying a lossless compression of frequent factor-sets*, and (2) *identifying a set of optimal rules*. A lossless compression of frequent factor-sets represents a small basic set of factor-sets containing the complete information regarding to its corresponding frequent factor-sets. An optimal rule is a general rule that has minimal antecedent and maximal consequent and presents the most general knowledge which is not redundant. To facilitate user interpretation, it is important to find a set of optimal rules to pruning redundant ones. The resulting set size is reduced by removing a rule if it conveys the same or less general information than another rule of the same usefulness and relevance.

4 Non-redundant Reclassification Rules and Algorithm

It is well known that the set of closed frequent item-sets [6] is sufficient to drastically reduce the association rule set without information loss. Similar approach is proposed for reclassification rule mining in this paper.

By a closed factor-set χ in S we mean a term

$\chi = [(a_1, \alpha_1 \rightarrow \beta_1) \wedge (a_2, \alpha_2 \rightarrow \beta_2) \wedge \ldots \wedge (a_p, \alpha_p \rightarrow \beta_p)]$ *iff none of its supersets* χ' satisfies $supL(\chi) = supL(\chi')$, $supR(\chi) = supR(\chi')$.

It means that χ is not closed if at least one of its immediate supersets has both supports the same as χ. Referring back to the example, the factor-set [(b, 0→2) (d, 2→1)(g, 0→1)] has left support count 0.4 and right support count 0.4. Notice that the supports of [(b, 0→2) (d, 2→1)] are identical to [(b, 0→2) (d, 2→1)(g, 0→1)] so [(b, 0→2) (d, 2→1)] should not be considered as a closed factor-set. On the other hand, [(b, 0→2) (d, 2→1)(g, 0→1)] is a closed factor-set because it does not have the same support counts as any of its supersets.

By a frequent closed factor-sets ξ, we mean that

$\xi = [(a_1, \alpha_1 \rightarrow \beta_1) \wedge (a_2, \alpha_2 \rightarrow \beta_2) \wedge \ldots \wedge (a_p, \alpha_p \rightarrow \beta_p)]$ is a frequent closed factor-set if ξ is a closed factor-set, $supL(\xi) \geq s_1$ and $supR(\xi) \geq s_2$.

Referring back to the example, the predefined thresholds for left support is 20% and 40% for right support, [(b,0→2) (d,2→1)(g,0→1)] is a frequent closed factor-set because its left support is 40% and right support is 40%. As compared with previous approaches to all frequent factor-sets, the set of frequent closed factor-sets is a loss-less compression version of it. In this example, we have eight frequent factor-sets (Table 2) and only two of them are frequent closed factor-sets (Table 3).

Table 3. A set of Frequent Closed Factor-sets

supL	***supR***	**Frequent Closed Factor-sets**
0.40	0.40	[(b,0→2)(d,2→1)(g,0→1)], (b,2→0)(d,1→2)(g,1→0)]

Let us first formally define what we mean by a redundant reclassification rules. Let r_i denotes the rule $[(X,\alpha\rightarrow\beta)_i \Rightarrow (Y,\phi\rightarrow\psi)_i]$. r_1 is more general than a rule r_2 if r_2 is valid, r_1 and r_2 have the same supports and confidence, $(X,\alpha\rightarrow\beta)_1 \subseteq (X,\alpha\rightarrow\beta)_{2,}$ and $(Y,\phi\rightarrow\psi)_1 \subseteq (Y,\phi\rightarrow\psi)_{2,}$ denoted as $r_1 \leq r_2$. Let R = $\{ r_1, r_2, \ldots, r_n \}$ be a set of rules and all of them have the same left and right support counts and confidence, i.e., $supL(r_1) = supL(r_2) = \ldots = supL(r_n)$, $supR(r_1) = supR(r_2) = \ldots = supR(r_n)$, and $conf(r_1) = conf(r_2) = \ldots = conf(r_n)$. r_m is a redundant rule if r_n is valid and

- $r_n \leq r_m$,
- $(Y,\phi\rightarrow\psi)_n = (Y,\phi\rightarrow\psi)_m$ and $(X,\alpha\rightarrow\beta)_n \Rightarrow (X,\alpha\rightarrow\beta)_m$, but not $(X,\alpha\rightarrow\beta)_m \Rightarrow (X,\alpha\rightarrow\beta)_n$, or
- $(X,\alpha\rightarrow\beta)_n = (X,\alpha\rightarrow\beta)_m$ and $(Y,\phi\rightarrow\psi)_n \Rightarrow (Y,\phi\rightarrow\psi)_m$.

Therefore, the non-redundant rules are those that are having shortest antecedent and longest consequent. A non-redundant reclassification rule *NR* is defined as

> $\mathcal{NR}$: $[(X,\alpha\rightarrow\beta) \Rightarrow (Y,\phi\rightarrow\psi)]$ *is not redundant iff none of the existed rules* $\mathcal{R}'$: $[(X',\alpha'\rightarrow\beta') \Rightarrow (Y',\phi'\rightarrow\psi')]$ *with* $supL(\mathcal{NR}) = supL(\mathcal{R}')$, $supR(\mathcal{NR}) = supR(\mathcal{R}')$, $conf(\mathcal{NR}) = conf(\mathcal{R}')$, satisfies $(X',\alpha'\rightarrow\beta') \subseteq (X, \alpha\rightarrow\beta)$ *and* $(Y,\phi\rightarrow\psi) \subseteq (Y',\phi'\rightarrow\psi')$.

Referring back to the example, notice that both the supports and confidence of $R_{1,}$ (d, 2→1) ⇒ (b, 0→2), and R_5, (d, 2→1) ⇒ (b, 2→0) (g, 0→1), are identical. The antecedent in both rules is the same element (d, 2→1) and the consequent of R_5 is one factor longer than R_1. Rule R_1 should not be considered as a non-redundant reclassification rule. On the other hand, R_5 is a non-redundant reclassification rule because it has the shortest length of its premise and the longest length of consequent and does not have the same support counts and confidence as any of its super rules.

StrategyGeneratorIII consists of two steps: mining frequent closed factor-sets and their corresponding rules. The new algorithm differs from *StrategyGeneratorII* in finding a set of closed factor-sets for rule generation. The pseudo-code for generating frequent closed factor-sets is listed in Fig. 2.

Referring back to our previous example, a complete set of non-redundant reclassification rules are constructed based on the whole set of frequent closed factor-sets. Only four such rules are generated in this new method and listed in Table 4. As

```
$F_{max}$:= get all frequent factor-sets f with max Length from frequentFactorQueue
maxLength:= the length of a factor-set in $F_{max}$
CompactFactorQueue := $\cup F_k$
for (i := (maxLength-1), i- -, i:=1) do
    $F_i$:= get all frequent factor-sets f' with length i from frequentFactorQueue
    for each f' in $F_i$ do
        if f'.supL > max($\forall$ f.supL in $F_{i+1}$ )
            if f'.supR > max($\forall$ f.supR in $F_{i+1}$ )
                CompactFactorQueue := $\cup$ f'
            end if
        end if
    end for
end for
```

Fig. 2. Algorithm for generating closed frequent factor-sets

compared with *StrategyGeneratorII* algorithm, it generates a relatively small set of non-redundant reclassification rules instead of all valid reclassification rules. In this example, we have eight valid reclassification rules and four of them are redundant. The size of the resulting set is reduced by about 50%. And, all of the rules have shortest antecedent and longest consequent. Thus, it is easier for the user to explore the knowledge space and efficiently interpret the result.

Table 4. A set of non-redundant Reclassification Rules

Rule	*supL*	*supR*	*Conf.*
R_5: (d, 2→1) ⇒ (b, 2→0) (g, 0→1)	2/5	2/5	100%
R_6: (b, 0→2) (g, 0→1) ⇒ (d, 2→1)	2/5	2/5	100%
R_7: (d, 1→2) ⇒ (b, 2→0) (g, 1→0)	2/5	2/5	100%
R_8: (b, 2→0) (g, 1→0) ⇒ (d, 1→2)	2/5	2/5	100%

5 Conclusion

The standard data mining focuses on patterns that summarize the data, and as a result, such patterns are required to be further processed in order to respond opportunities of action. To address this problem, mining reclassification rules provides a way to extract relevantly useful, understandable, and workable strategies by comparing the profiles of two sets of relevant objects. In this paper, we presented the notions of closed factor-sets, non-redundant reclassification rules, and introduced algorithm *StrategyGeneratorIII*. This new algorithm based on the concept of frequent closed factor-sets to generate a very concise set of rules. These discovered rules are non-redundant and bring the same information as the original set of all valid rules to the end-user. The number of rules produced by the new approach is fairly smaller than the rule set from the previous approach. Therefore, it would be easier for the user to comprehend the rule result in timely manner. There is a lot of interesting directions for future work. For example, we intend to generalize the set of reclassification rules even further by incorporating meta-mining techniques, also apply the approach to large variety of databases and application domains.

Acknowledgments. This research is partially supported by the Asian Studies Research Grant Program of the Center for International Studies at the University of Pittsburgh.

References

1. He, Z., Xu, X., Deng, S.: Mining Cluster-Defining Actionable Rules. In: Proceedings of NDBC 2004 (2004)
2. He, Z., Xu, X., Deng, S., Ma, R.: Mining Action Rules From Scratch. Expert Systems with Applications 29(3), 691–699 (2005)
3. Im, S., Raś, Z.W.: Action Rule Extraction from a Decision Table: ARED. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) Foundations of Intelligent Systems. LNCS, vol. 4994, pp. 160–168. Springer, Heidelberg (2008)
4. Ling, C.X., Chen, T., Yang, Q., Chen, J.: Mining Optimal Actions for Intelligent CRM. In: 2002 IEEE International Conference on Data Mining, pp. 767–770. IEEE Computer Society, Maebashi City (2002)
5. Liu, B., Hsu, W., Ma, Y.: Identifying Non-actionable Association Rules. In: Proceedings of KDD 2001, San Francisco, CA, USA, pp. 329–334 (2001)
6. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Discovering Frequent Closed Itemsets for Association Rules. In: Beeri, C., Bruneman, P. (eds.) ICDT 1999. LNCS, vol. 1540, pp. 398–416. Springer, Heidelberg (1998)
7. Pawlak, Z.: Information Systems - Theoretical Foundations. Information Systems Journal 6, 205–218 (1981)
8. Piatesky-Shapiro, G., Matheus, C.J.: The interestingness of deviations. In: Proceedings of AAA Workshop on Knowledge Discovery in Database, pp. 25–36. AAAI Press, Menlo Park (1994)
9. Raś, Z.W., Tsay, L.-S.: Discovering Extended Action-Rules (System DEAR). In: Proceedings of the IIS 2003 Symposium, Advances in Soft Computing, pp. 293–300. Springer, Zakopane (2003)
10. Raś, Z.W., Wieczorkowska, A.: Action-Rules: How to Increase Profit of a Company. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS, vol. 1910, pp. 587–592. Springer, Heidelberg (2000)
11. Tsay, L.-S., Raś, Z.W.: Action Rules Discovery: System DEAR2, Method and Experiments. Journal of Experimental and Theoretical Artificial Intelligence 17(1-2), 119–128 (2005)
12. Tsay, L.-S., Raś, Z.W.: E-Action Rules. In: Lin, T.Y., Xie, Y., Wasilewska, A., Liau, C.-J. (eds.) Foundations of Data Mining. Studies in Computational Intelligence, pp. 261–272. Springer, Heidelberg (2007)
13. Tsay, L.-S., Raś, Z.W.: Discovering the Concise Set of Actionable Patterns. In: An, A., Matwin, S., Raś, Z.W., Ślęzak, D. (eds.) Foundations of Intelligent Systems. LNCS (LNAI), vol. 4994, pp. 169–178. Springer, Heidelberg (2008)
14. Tsay, L.-S., Ras, Z.W., Im, S.: Reclassification Rules. In: IEEE/ICDM Workshop on Foundations of Data Mining (FDM 2008), pp. 619–627. IEEE Computer Society, Pisa (2008)

15. Wong, R.C.-W., Fu, A.W.-C.: ISM: Item Selection for Marketing with Cross-selling Considerations. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS, vol. 3056, pp. 431–440. Springer, Heidelberg (2004)
16. Yang, Q., Yin, J., Lin, C.X., Chen, T.: Postprocessing Decision Trees to Extract Actionable Knowledge. In: Proceedings of ICDM 2003, pp. 685–688. IEEE Computer Society, Florida (2003)
17. Zhang, H., Zhao, Y., Cao, L., Zhang, C.: Combined Association Rule Mining. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS, vol. 5012, pp. 1069–1074. Springer, Heidelberg (2008)

A Hash Trie Filter Approach to Approximate String Matching for Genomic Databases

Ye-In Chang, Jiun-Rung Chen, and Min-Tze Hsu

Dept. of Computer Science and Engineering,
National Sun Yat-Sen University, Kaohsiung, Taiwan
changyi@cse.nsysu.edu.tw

Abstract. For genomic databases, approximate string matching with k errors is often considered for genomic sequences, where k errors could be caused by substitution, insertion, or deletion operations. In this paper, we propose a new approximate string matching method, the *hash trie filter*, for efficiently searching in genomic databases. Our method not only reduces the number of candidates by pruning some unreasonable matched positions, but also dynamically decides the number of ordered matched grams of one candidate, which results in the increase of precision. The experiment results show that the hash trie filter outperforms the well-known $(k + s)$ q-samples filter in terms of the response time and the precision, under different lengths of the query patterns and different error levels.

Keywords: Approximate string matching, filter, hash trie.

1 Introduction

Bioinformatics has received increased publicity over the past few years, in large part due to its importance to the Human Genome Project. With the increment of genomic sequences, numerous large databases hold these DNA and protein sequences [3]. DNA sequences could be very long. For example, the human genome contains about 3×10^9 base pairs (bps). Because of the long length of each genomic sequence and the increase of the size of genomic sequence databases, the importance of efficient searching methods for fast queries grows.

Approximate string matching (ASM) is often applied when searching in the databases, where ASM finds the positions of all subsequences in the text that allows at most k edit operations to make a substring of the text become the query pattern. In the DNA related research, *mutations* occur very often, where a mutation is defined as a heritable change, *i.e.*, replacement, insertion, or deletion, in the DNA sequence, caused by a faulty replication process [2]. The problem of finding these mutations could be solved by ASM, since a DNA sequence could be thought as a traditional string consisted of characters {A, C, G, T} in the string matching problem.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 816–825, 2009.

There are several principal database search methods used to compute pairwise comparisons between a query sequence and each of the sequences stored within a database, *e.g.*, Smith-Waterman [7], FASTA [5], and BLAST [1]. However, these methods may run slow and need large amounts of memory [6]. On the other hand, *filter* methods are currently very active for approximate string matching. It is based on finding fast methods to discard large areas of the text that cannot match, and applying another method to verify the candidates, *e.g.*, the simple dynamic programming method. Most of filter methods deal with q-grams of the pattern, *i.e.*, continuous substrings with length q. The idea is that whenever an approximate match occurs, it has to resemble the original pattern. This resemble is reflected by the existence of the same q-grams both in the pattern and in its approximate match [8]. In practice, filter methods are the fastest ones [4].

According to the consideration of the local/global order of subpatterns in a sequence, filter methods could be classified into two types: one ensures the local order of the characters within a gram or subpattern, and one ensures the global order of grams or subpatterns within a sequence. Figure 1 shows the concept of the global order. Most of those previous filter methods consider only the local order. So far, only the $(k + s)$ q-samples filter [8,9] considers the global order, which could improve the precision. However, it still has some disadvantages. The $(k + s)$ q-samples filter cannot handle the situations when the length of a query pattern is short. Moreover, when the length of a query pattern is too long, it will lead to a large value of h according to its formula, where h is the distance of sampling. However, since the length of a sample is fixed to $q = \log_4 |P|$, the large non-sampled area within h would be neglected, which reduces the precision.

To improve the $(k+s)$ q-samples filter, we propose a new method, the *hash trie filter*. We will construct a hash trie for indexing the database text, X, in advance. Then, when a user inquires a query, we utilize several techniques to efficiently find the possible candidates in text X, which not only ensure that the global order of passed subpatterns in a candidate is correct, but also dynamically decide the number of matched subpatterns of a candidate to increase the precision. From our performance study, we show that the hash trie filter could provide better performance than the $(k + s)$ q-samples filter in terms of the response time and precision, since the former generates less number of candidates than the latter.

The rest of the thesis is organized as follows. Section 2 briefly describes the $(k + s)$ q-samples filter [8,9]. Section 3 presents the proposed hash trie filter. In Sect. 4, we compare the performance of the hash trie filter with that of the $(k + s)$ q-samples filter. Finally, Sect. 5 states the conclusion.

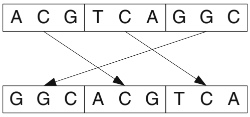


Fig. 1. An example of different global order of grams between two sequences

2 The $(k + s)$ q-Samples Filter

Sutinen and Tarhio proposed several filter methods for approximate string matching [8,9]. These methods focus on $(k + s)$ consecutive q-samples, and thus we call these methods "the $(k + s)$ q-samples filter" in this paper. This method utilizes the idea of sampling for the text, X. The distance of sampling is $h = \lfloor (m-k-q+1)/(k+s) \rfloor$, and the last q-gram within the sampling interval is examined as a q-sample. The reason for this sampling is that to ensure s pieces matched, the pattern needs to be cut into $(k + s)$ pieces. At the search time, instead of one single q-sample, they consider text windows of consecutive sequences of $(k + s)$ q-samples. Each of these q-samples is searched. If at least s q-samples are found, the area of $X[(j-1)-m'h-2k-q+2, (j-1)+m-(m'-1)h+k-q]$ forms a candidate. Note that in the part of concluding remarks of [8], the authors mentioned that they had tried simulating their method for $s > 2$, but the total performance does not increase significantly. Therefore, in fact, the authors only focused on $(k + 2)$ consecutive samples, *i.e.*, $s = 2$.

3 A Hash Trie Filter Approach for ASM

In this section, first, we present a data structure, the hash trie, for efficiently access the q-grams split from sequences of genomic database. Next, we present a new query processing method for ASM.

3.1 The Construction of a Hash Trie

The hash trie is built for indexing text sequences in a database. For the given text, X, we traverse its q-grams in the hash trie (HT) one character by one character, and store the occurring positions of each gram under the traversing path. For example, for 3-grams (*i.e.*, $q = 3$) split from text X shown in Fig. 2-(a), Fig. 2-(b) shows the resulting hash trie. The length of a gram used in the hash trie is q, and it is determined by the formula $q = \lceil \log_\sigma n \rceil$ [8], where σ is the size of the alphabet and n is the length of X. Note that we let $\sigma = 4$ for the DNA sequences. Based on this formula, the height of a hash trie, q, is proportional to the logarithm of n. To avoid missing any information, the last $(q - 1)$ characters of X still have to be split, although their length is less than q. The breadth of a node of the hash trie is $\sigma = 4$, since the alphabet is $\Sigma = \{\text{A,C,G,T}\}$. Therefore, the hash trie has 4^q leave nodes. Initially, the hash trie is a complete trie.

3.2 Query and Search

After constructing the hash trie, HT, for the sequence stored in the genomic database, the user can inquire a query on it. In our query processing method, there are 5 main steps described as follows.

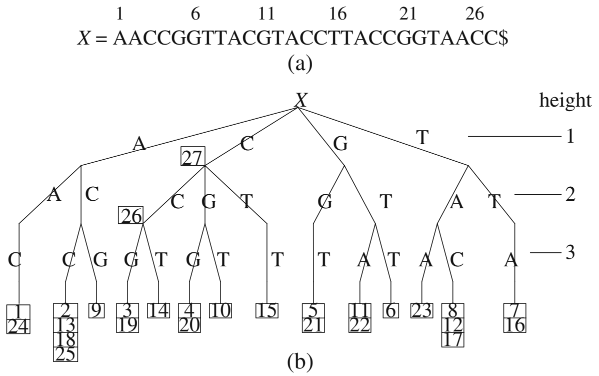


Fig. 2. The hash trie with $q = 3$ in our example: (a) the given text, X; (b) the related hash trie

Step 1: Deciding Length q' of a Subpattern. In the hash trie filter, when the query is inquired, first, we compute length q' of a subpattern by using the formula of the $(k + s)$ q-samples filter, $q' = \lceil \log_4 m \rceil$, where m is the length of query pattern P. Note that since we let $q = \lceil \log_4 n \rceil$ for text X $(n = |X|)$ and $m \leq n$, the condition $q' \leq q$ is always satisfied. In this way, the subpattern with length equal to q' will always be found, if it exists in the hash trie. Then, our method will dynamically decide the number of exactly matching subpatterns (s) which satisfy the global order, according to the number of allowed errors (k). We let $NumSP = \lfloor m/q' \rfloor$ and $s = NumSP - k$. When $NumSP$ becomes larger or k becomes smaller, s will be larger, and the precision will be increased, too. If s is smaller than a threshold, w, it means that the precision is lower than a level determined by heuristic experiments. For this case, we will reduce the length of one gram by 1 to increase $NumSP$.

Step 2: Constructing Table *PT* for the Subpatterns. In this step, we split query pattern P into several non-overlapped grams, and traverse them in the hash trie, HT. Moreover, we generate a position table (PT) to store the positions of each subpattern P_i of P. Figure 3 shows an example of how query pattern P is split into non-overlapped grams. Assume that m is 17 and k is 3. Since the length of a gram, q', is equal to $\lceil \log_4 17 \rceil = 3$, we split query pattern P into 5 $(= NumSP)$ non-overlapped grams with length 3, *i.e.*, $\{P_1, P_2, P_3, P_4, P_5\}$. The remaining part whose length is less than q' ("GG" in this example) is ignored at this point, and will be considered in Step 4. Moreover, its length is stored in $lastLen$. In this example, the value of $lastLen$ is 2.

After the query pattern, P, is split into several non-overlapped grams, we traverse hash trie HT to find the positions of each gram of query pattern P occurring in text X. For each subpattern P_i in the previous example shown in Fig. 3, Fig. 4 shows the related table PT for storing their positions, according to the hash trie shown in Fig. 2-(b).

Step 3: The Pruning Step. In this step, we will prune those unreasonable start/end positions in table PT. Note that a start or end position denoted by

$$P = \underline{\text{CCG}}\,\underline{\text{GTT}}\,\underline{\text{ACG}}\,\underline{\text{CCT}}\,\underline{\text{TAC}}\text{GG\$}$$

P	CCG	GTT	ACG	CCT	TAC
P_i	P_1	P_2	P_3	P_4	P_5

Fig. 3. Splitting query pattern P into 3-grams (subpatterns)

$z1$ or $z2$, respectively, is unreasonable, if (1) for the start position $z1$, we have $z1 < (1-k)$; or (2) for the end position $z2$, we have $z2 > (n+k)$.

Let PL_i $(= PT[P_i])$ denote a list of positions stored in table PT, which indicates the positions of subpattern P_i occurring in text X. Note that we allow at most k differences between query pattern P and text X. Assume that all k errors occur in front of each position of PL_i, which results in the smallest possible start position. Then, the possible start position, $z1$, of the first gram, where the related subpattern P_i matches a substring of text X, is $z1 = PL_i[j] - \sum_{r=1}^{i-1} |P_r|$, where $j \in 1 \ldots |PL_i|$. The value of $z1$ should not be smaller than $(1-k)$; otherwise, this substring of text X can not become query P within k edit operations. Therefore, we can prune those unreasonable positions. In the previous example shown in Fig. 4, position 8 of subpattern P_5 will be pruned, since we have $z1 = PL_5[1] - 12 = 4 - 12 = -8 < (1-k) = -2$.

Let $PL_i(= PT[P_i])$ denote a list of positions stored in table PT. Assume that all k errors occur after each position of PL_i, which results in the largest possible end position. The possible end position, $z2$, of a substring of text X, will be $z2 = PL_i[j] + \sum_{r=i}^{NumSP} |P_r| - 1 + k = PL_i[j] + (q' * (NumSP - (i-1)) - 1) + k$, where $j \in 1 \ldots |PL_i|$, and "-1" is due to the start index = 1. The value of $z2$ should not be larger than $(n+k)$; otherwise, this substring of text X can not become query P within k edit operations. In the previous example shown in Fig. 4, position 19 of subpattern P_1 is pruned, since its end position is $z2 = PL_1[j] + (3 * 5 - 1) + 3 = 19 + 17 = 36 > (n+k) = 30$.

After pruning all unreasonable positions in table PT, we store the reasonable end positions for the rest of positions of table PT into a new array, RM. Figure 5 shows an example of array RM, where Fig. 5-(a) shows those reasonable end positions for the previous example shown in Fig. 4, and Fig. 5-(b) shows the corresponding array RM. Initially, the value of each cell in array RM is set to 0. The value of one cell will be set to 1 according those reasonable end positions of P_i. For example, the reasonable end positions of P_5 are 17 and 22. Therefore, the corresponding cells 17 and 22 of P_5 are set to 1, as shown in Fig. 5-(b).

PT

gram	positions
P_1	(3 , 19)
P_2	(6)
P_3	(9)
P_4	(14)
P_5	(8, 12 , 17)

Fig. 4. Table PT for storing the positions of subpattern P_i in text X

gram	related end
P_1	(20)
P_2	(20)
P_3	(20)
P_4	(22)
P_5	(17 , 22)

(a)

RM

P_i	17	18	19	20	21	22
P_1	0	0	0	1	0	0
P_2	0	0	0	1	0	0
P_3	0	0	0	1	0	0
P_4	0	0	0	0	0	1
P_5	1	0	0	0	0	1

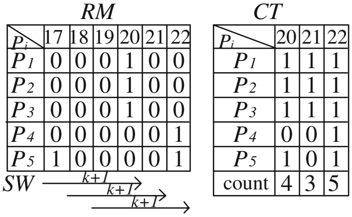
(b)

Fig. 5. An example of array RM: (a) the original reasonable end positions; (b) the corresponding array RM

Step 4: Finding Ordered Subpatterns. For a substring of text X, when there are at least s subpatterns P_i of query pattern P matched with this substring, it will be a possible candidate which also satisfies the global order. The reason is that k errors will affect at most k subpatterns in the worst case. For example, assume that the query is split into $NumSP = 5$ subpatterns, and k is 3. For a candidate, since 3 errors affect at most 3 subpatterns, there should exist at least $s = (NumSP - k) = (5 - 3) = 2$ subpatterns matched. To achieve this goal, we utilize array RM and a sliding window with size $(k+1)$ to find the ordered s subpatterns of a candidate.

We use a sliding window SW with size $(k+1)$, and a counting table CT with size $(maxEnd - minEnd - 1 - k) * NumSP$, to find the ordered s subpatterns of a candidate, where $maxEnd$ and $minEnd$ are the minimal and the maximal reasonable end positions, respectively. Figure 6 shows an example of table CT, where $CT[P_i, j]$ is used to count the number of occurrence of subpattern P_i between $RM[P_i, (j-k)]$ and $RM[P_i, j]$, *i.e.*, a sliding window with size $(k+1)$. To efficiently count these numbers, for the first sliding window, we use k times of "+" operations on the columns of table RM between $RM[P_i, minEnd]$ and $RM[P_i, (k+minEnd)]$, and store the result in $CT[P_i, (k+minEnd)]$. Therefore, $CT[P_i, (k + minEnd)] = RM[P_i, minEnd] + \ldots + RM[P_i, (k + minEnd)]$. The rest of values of table CT, except for the first sliding window, is calculated as $CT[P_i, j] = CT[P_i, (j-1)] + RM[P_i, j] - RM[P_i, (j-k-1)]$, where $RM[P_i, j]$ is the new column that enters the sliding window, and $RM[P_i, (j-k-1)]$ is the old column that leaves the sliding window (*i.e.*, shifting the sliding window to right by one column in table RM).

In table CT, there exists one more row, *count*, in the bottom of table CT. $CT[count, j]$ records the number of matched subpatterns in the current sliding window. $CT[count, j]$ is increased by 1, whenever $CT[P_i, j] \neq 0$ for $i = 0 \ldots NumSP$. As we mentioned previously, if $CT[count, j]$ is not less than s, it indicates that there exists a candidate with $CT[count, j]$ ordered and matched subpatterns. Take table CT shown in Fig. 6 as an example. Assume that $s = (NumSP - k) = (5 - 3) = 2$. Since $CT[count, 20] = 4$, there are 4 ($> s$) subpatterns occurring in order, which means that there may be an ASM happened. Therefore, we mark the range related to position 20 to represent a candidate, *i.e.*, $X[((20 + lastLen) - (m + 2k)), (20 + lastLen)] = X[((20 + 2) - (17 + 2 * 3)), (20 + 2)] = X[-1, 22]$, where $lastLen$ is the length of the rest of query



RM

P_i	17	18	19	20	21	22
P_1	0	0	0	1	0	0
P_2	0	0	0	1	0	0
P_3	0	0	0	1	0	0
P_4	0	0	0	0	0	1
P_5	1	0	0	0	0	1

CT

P_i	20	21	22
P_1	1	1	1
P_2	1	1	1
P_3	1	1	1
P_4	0	0	1
P_5	1	0	1
count	4	3	5

Fig. 6. The result of finding s ordered subpatterns of query pattern P in text X

pattern P that do not be split, as described in Step 2. Because the minimum start index of X is 1, $X[-1, 22]$ is modified to $X[1, 22]$. This range of text X (*i.e.*, a candidate) and query pattern P will be verified with dynamic programming in Step 5.

The length of a candidate in our hash trie filter is $(m + 2k)$, while it is about $(m + 3k)$ in the $(k + s)$ q-samples method [8]. We use Fig. 7 to illustrate why this length could be $(m + 2k)$, where this figure shows three kinds of situations that should be passed to be examined. Figure 7-(a) is an example for exact matching. Figure 7-(b) is an example for ASM, where k errors are all insertions. Figure 7-(c) is an example for ASM, where k errors are all deletions. From these examples, we know that we can cover the subpatterns $P_1 \ldots P_{NumSP}$ by only passing a substring with length $(m + 2k)$. Note that we have to consider the length $(lastLen)$ of the characters which is not long enough to be a subpattern with length q'. That is, the end position of the candidate is $(CT[j] + lastLen)$, where $CT[j]$ is the position with a number of matched subpatterns larger than or equal to s. Therefore, the range of a candidate would be $X[(CT[j] + lastLen) - (m + 2k), (CT[j] + lastLen)]$. The length of a candidate will effect the time for verifying. Therefore, the proposed hash trie filter is more precise and more efficient than the $(k + s)$ q-samples filter.

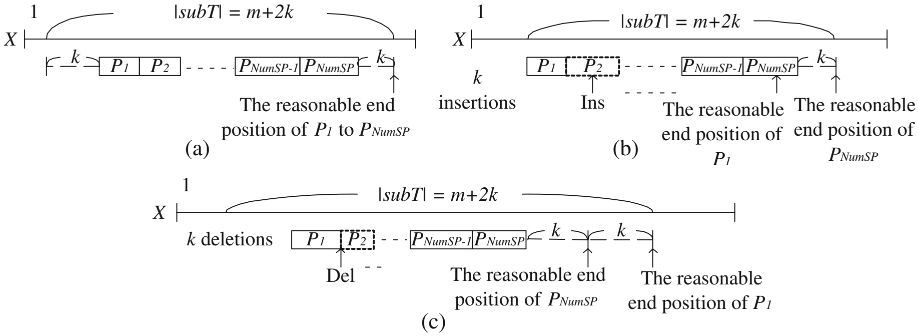

1
|subT| = m+2k
X
k
P1
P2
PNumSP-1
PNumSP
k
The reasonable end position of P1 to PNumSP
(a)
1
|subT| = m+2k
X
k insertions
P1
P2
PNumSP-1
PNumSP
k
Ins
The reasonable end position of P1
The reasonable end position of PNumSP
(b)
1
|subT| = m+2k
X
k deletions
P1
P2
PNumSP-1
PNumSP
k
k
Del
The reasonable end position of PNumSP
The reasonable end position of P1
(c)

Fig. 7. Three kinds of situations: (a) an exact match; (b) an ASM where all errors are insertions (Ins); (c) an ASM where all errors are deletions (Del)

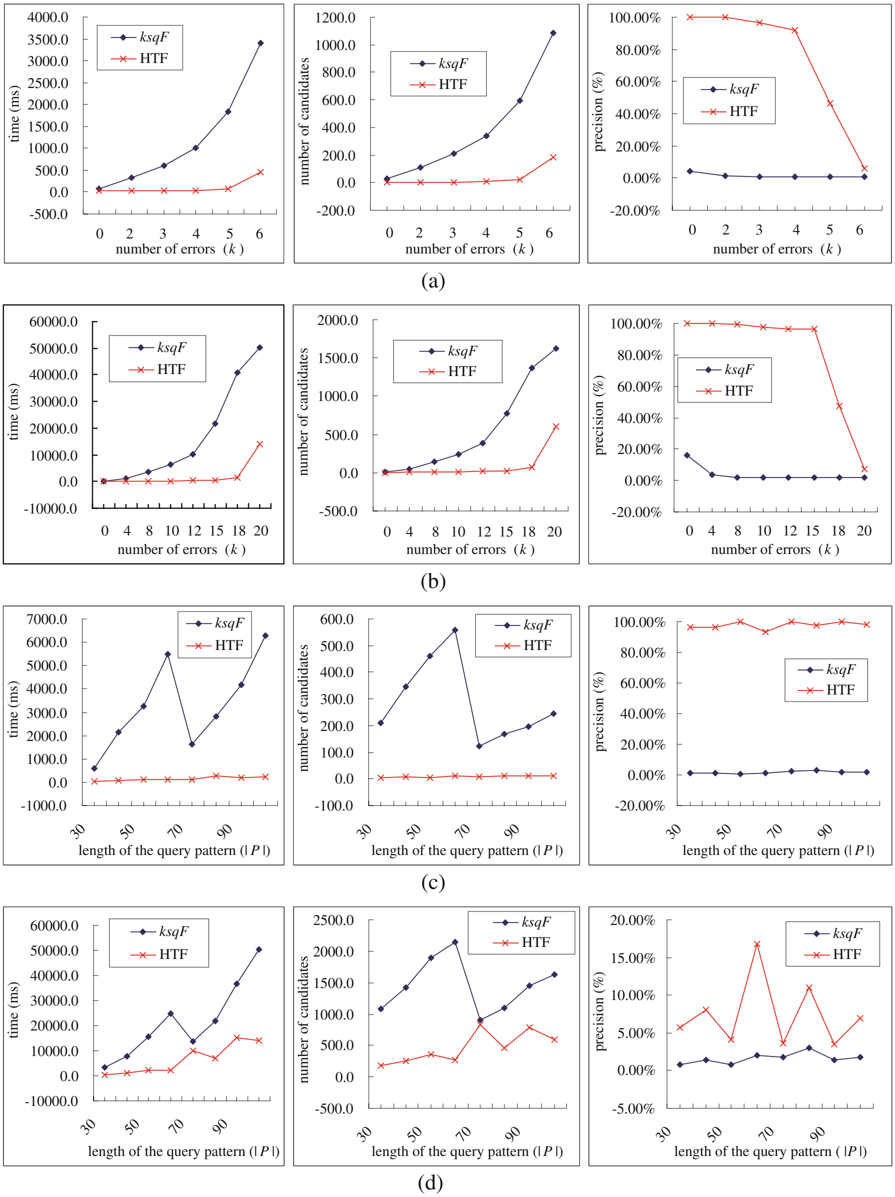
Step 5: Verification with Dynamic Programming. The last step is to verify the candidates with the query by applying dynamic programming. This step could be done by any existing dynamic programming approach, *e.g.*, [7].

4 Performance

In this section, we study the performance of the proposed hash trie filter, and make a comparison with the $(k + s)$ q-samples filter by simulation. The same query patterns and dynamic programming are used for both of the hash trie filter (HTF) and the $(k + s)$ q-samples filter ($ksqF$). We chose the DNA sequence of gene "$Z73521$" contained in human chromosome 16, whose length is 7881 base pairs, from GenBank (http://www.ncbi.nlm.nih.gov) to be our database sequence used in the performance study. The query sequences with lengths from 10 to 100 were generated randomly from the DNA sequence. Generally, the optimal error level for the filtration methods seems to be about $0 - 20\%$ [8]. Therefore, our experiments were based on the error levels about $0 - 20\%$.

Figure 8 shows the comparisons of the response time, the number of processed candidates, and the precision, between the $(k + s)$ q-samples filter and the hash trie filter. Figure 8-(a) shows the comparisons under short query patterns with length 30 and varying the numbers of errors. Figure 8-(b) shows the comparisons under long query patterns with length 100 and varying the numbers of errors. Figure 8-(c) shows the comparisons under a low error level, $(m/k) = 10\%$, where the lengths of the query sequences are from 30 to 100. Figure 8-(d) shows the comparisons under a high error level, $(m/k) = 20\%$, where the lengths of the query sequences are from 30 to 100. From these results, we show that the hash trie filter needs less response time than the $(k + s)$ q-samples filter under those above conditions. The reason is that the number of candidates of the hash trie filter is fewer than that of the $(k + s)$ h-samples filter. Therefore, the verification time of the hash trie filter is also less than that of the $(k + s)$ q-samples filter.

From Fig. 8, we also show that the precision of the hash trie filter is higher than the $(k + s)$ q-samples filter. The precision is measured by the formula: $(N_{\text{hit}}/N_{\text{candidates}}) * 100\%$, where N_{hit} is the number of hitting answers within the candidates, and $N_{\text{candidate}}$ is the number of candidates. Since the number of candidates of the hash trie filter is fewer than that of the $(k + s)$ q-samples filter, the precision of the hash trie filter will be higher than that of the $(k + s)$ q-samples filter. Obviously, as the number of errors increases, the precision of both methods decreases. The reason for the high precision of the hash trie filter is due to the large value of s. For example, when $m = 30$ and $k = 2$, the value of s in the hash trie filter will be 8. Although our strategy makes use of the large value of s to determine a candidate, it will not cause any missing case. The reason is that for a candidate, k errors affect at most k subpatterns of this candidate, as we mentioned in the previous section.



(a)

(b)

(c)

(d)

Fig. 8. Comparisons of performance between the $(k+s)$ q-samples filter and the hash trie filter: (a) under short query patterns with length = 30; (b) under long query patterns with length = 100; (c) under a low error level, $(m/k) = 10\%$; (d) under a high error level, $(m/k) = 20\%$

5 Conclusion

Genomic sequence databases, *e.g.*, DNA sequence databases, are widely used by molecular biologists for searching approximate matches. In this paper, we have given the definition of a hash trie and proposed a new filtration method, the hash trie filter, for approximate string matching in the genomic sequences. Our method not only utilizes several techniques to reduce the number of candidates, but also efficiently finds the ordered subpatterns for each candidate, which will not cause any missing case. From the simulation results, we have shown that the hash trie filter could provide better performance than the $(k + s)$ q-samples filter in terms of the response time, the number of candidates, and the precision, under various conditions.

Acknowledgements. This research was supported in part by the National Science Council of Republic of China under Grant No. NSC-95-2221-E-110-079-MY2. The authors also like to thank "Aim for Top University Plan" project of NSYSU and Ministry of Education, Taiwan, for partially supporting the research.

References

1. Altschul, S., Gish, W., Miller, W., Myers, E., Lipman, D.: Basic Local Alignment Search Tool. Journal of Molecular Biology 215(3), 403–410 (1990)
2. Friedberg, E.C., Walker, G.C., Siede, W.: DNA Repair and Mutagenesis. American Society Microbiology (1995)
3. Houle, J.L., Cadigan, W., Henry, S., Pinnamaneni, A., Lundahl, S.: Database Mining in the Human Genome Initiative (2000), http://www.biodatabases.com/whitepaper01.html
4. Karkkainen, J., Na, J.C.: Faster Filters for Approximate String Matching. In: Proc. of Workshop on Algorithm Engineering and Experiments, pp. 1–7 (2007)
5. Lipman, D.J., Pearson, W.R.: Rapid and Sensitive Protein Similarity Searches. Science 227(4693), 1435–1441 (1985)
6. Ma, B., Tromp, J., Li, M.: PatternHunter: Faster and More Sensitive Homology Search. Bioinformatics 18(3), 440–445 (2002)
7. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. Journal of Molecular Biology 147(1), 195–197 (1995)
8. Sutinen, E., Tarhio, J.: On Using q-Gram Locations in Approximate String Matching. In: Proc. of the 3rd Annual European Symp. on Algorithms, pp. 327–340 (1995)
9. Sutinen, E., Tarhio, J.: Approximate String Matching with Ordered q-Grams. Nordic Journal of Computing 11(4), 321–343 (2004)

Interval Regression Analysis with Soft-Margin Reduced Support Vector Machine

Chia-Hui Huang[1],* and Han-Ying Kao[2]

[1] Department of Information Management, Kainan University
No.1 Kainan Road, Luzhu Shiang, Taoyuan 33857, Taiwan
leohuang@mail.knu.edu.tw

[2] Department of Computer and Information Science, National Dong Hwa University
No.123 Hua-Shi Road, Hualien 97063, Taiwan
teresak_hk@yahoo.com.tw

Abstract. The support vector machine (SVM) has shown to be an efficient approach for a variety of classification problems. It has also been widely used in pattern recognition, regression and distribution estimation for crisp data. However, there are three main problems while using SVM model: (1) Large-scale: when dealing with large-scale data sets, the solution by using SVM with nonlinear kernels may be difficult to find; (2) Unbalance: the number of samples from one class is much larger than the number of samples from other classes. It causes the excursion of separation margin; (3) Noises and Interaction: the distribution of data becomes hard to be described and the separation margin between classes becomes a "gray" zone. Under this circumstance, to develop an efficient method is necessary. Recently the reduced support vector machine (RSVM) was proposed as an alternative of the standard SVM. It has been proved more efficient than the traditional SVM in processing large-scaled data. In this paper we introduce the principle of RSVM to evaluate interval regression analysis. In addition the soft margin method is proposed to modify the excursion of separation margin and to be effective in the gray zone.

Keywords: Interval regression analysis, reduced support vector machine, soft margin.

1 Introduction

Since Tanaka et al. [15] introduced the fuzzy regression model with symmetric fuzzy parameters, the properties of fuzzy regression have been studied extensively by many researchers. A collection of recent studies to fuzzy regression analysis can be reached at [7].

Fuzzy regression model can be simplified to interval regression analysis which is considered as the simplest version of possibilistic regression analysis with interval coefficients. Some coefficients in interval linear regression model tend to become crisp because of the characteristic of linear programming (LP) [15]. To

* Corresponding author.

B.-C. Chien et al. (Eds.): IEA/AIE 2009, LNAI 5579, pp. 826–835, 2009.

alleviate the issue of LP, Tanaka and Lee [14] propose an interval regression analysis with a quadratic programming (QP) approach which gives more diverse spread coefficients than a LP one.

The support vector machine (SVM) has been widely used in pattern recognition, regression and distribution estimation for crisp data [1,2,3,9,11,12,13,16]. Recently, using SVM to solve the interval regression model becomes an alterative approach. Hong and Hwang introduce SVM for multivariate fuzzy regression analysis [4] and evaluate interval regression models with quadratic loss SVM [5]. Jeng et al. [6] develop a support vector interval regression networks (SVIRNs) based on both SVM and neural networks.

However, there are three main problems while using SVM model:

(1) Large-scale: when dealing with large-scale data sets, the solution by using SVM with nonlinear kernels may be difficult to find.
(2) Unbalance: the number of samples from one class is much larger than the number of samples from other classes. It causes the excursion of separation margin.
(3) Noises and Interaction: the distribution of data becomes hard to be described and the separation margin between classes becomes a “gray” zone.

Under this circumstance, to develop an efficient method is necessary. The reduced support vector machine (RSVM) has been proved more efficient than the traditional SVM in processing large-scaled data [8]. The main idea of RSVM is to reduce the number of support vectors by randomly selecting a subset of samples. In this paper we introduce the principle of RSVM to evaluate interval regression analysis. Additionally, to modify the excursion of separation margin and to be effective in the gray zone, the soft margin method is proposed.

This paper is organized as follows. Section 2 reviews interval regression analysis by QP approach unifying the possibility and necessity models. Section 3 briefly reviews the basis of the theory of RSVM. Section 4 proposes the soft margin method and the formulation of RSVM in evaluating the interval regression models. Finally, section 5 gives the concluding remarks.

2 Interval Regression Analysis with QP Approach

In this section we review interval regression analysis by QP approach unifying the possibility and necessity models proposed by Tanaka and Lee [14].

An interval linear regression model is described as

$$Y(\boldsymbol{x}_j) = A_0 + A_1 x_{1j} + \ldots + A_n x_{nj} \tag{1}$$

where $Y(\boldsymbol{x}_j)$, $j = 1, 2, \ldots, q$ is the estimated interval corresponding to the real input vector $\boldsymbol{x}_j = (x_{1j}, x_{2j}, \ldots, x_{nj})^t$. An interval coefficient A_i is defined as (a_i, c_i), where a_i is the center and c_i is the radius. Hence, A_i can also be represented as

$$A_i = [a_i - c_i, a_i + c_i] = \{a_i - c_i \leq a \leq a_i + c_i\} \tag{2}$$

The interval linear regression model (1) can also be expressed as

$$\begin{aligned} Y(\boldsymbol{x}_j) &= A_0 + A_1 x_{1j} + \ldots + A_n x_{nj} \\ &= (a_0, c_0) + (a_1, c_1) x_{1j} + \ldots + (a_n, c_n) x_{nj} \\ &= (a_0 + \sum_{i=1}^{n} a_i x_{ij}, c_0 + \sum_{i=1}^{n} c_i |x_{ij}|) \end{aligned} \tag{3}$$

For a data set with crisp inputs and interval outputs, two interval regression models, the possibility and necessity models, are considered. By assumption, the center coefficients of the possibility regression model and the necessity regression model are the same [14]. For this data set the possibility and necessity estimation models are defined as

$$Y^*(\boldsymbol{x}_j) = A_0^* + A_1^* x_{1j} + \ldots + A_n^* x_{nj} \tag{4}$$

$$Y_*(\boldsymbol{x}_j) = A_{0*} + A_{1*} x_{1j} + \ldots + A_{n*} x_{nj} \tag{5}$$

where the interval coefficients A_i^* and A_{i*} are defined as $A_i^* = (a_i^*, c_i^*)$ and $A_{i*} = (a_{i*}, c_{i*})$, respectively. The interval $Y^*(\boldsymbol{x}_j)$ estimated by the possibility model must include the observed interval Y_j and the interval $Y_*(\boldsymbol{x}_j)$ estimated by the necessity model must be included in the observed interval Y_j. The following inclusion relations exist

$$Y_*(\boldsymbol{x}_j) \subseteq Y_j \subseteq Y^*(\boldsymbol{x}_j) \tag{6}$$

The interval coefficients A_i^* and A_{i*} can be denoted as

$$A_i^* = (a_i, c_i + d_i) \tag{7}$$

$$A_{i*} = (a_i, c_i) \tag{8}$$

which satisfy the inclusion relation $A_{i*} \subseteq A_i^*$ where c_i and d_i are assumed to be positive. Thus, the possibility model $Y^*(\boldsymbol{x}_j)$ and the necessity model $Y_*(\boldsymbol{x}_j)$ can also be expressed as

$$Y^*(\boldsymbol{x}_j) = (a_0 + \sum_{i=1}^{n} a_i x_{ij}, c_0 + \sum_{i=1}^{n} c_i |x_{ij}| + d_0 + \sum_{i=1}^{n} d_i |x_{ij}|) \tag{9}$$

$$Y_*(\boldsymbol{x}_j) = (a_0 + \sum_{i=1}^{n} a_i x_{ij}, c_0 + \sum_{i=1}^{n} c_i |x_{ij}|) \tag{10}$$

The interval regression analysis by QP approach unifying the possibility and necessity models subject to the inclusion relations can then be represented as follows

$$\begin{aligned} &\min \ \sum_{j=1}^{q} (d_0 + \sum_{i=1}^{n} d_i |x_{ij}|)^2 + \xi \sum_{i=0}^{n} (a_i^2 + c_i^2) \\ &s.t. \\ &\quad Y_*(\boldsymbol{x}_j) \subseteq Y_j \subseteq Y^*(\boldsymbol{x}_j), j = 1, 2, \ldots, p \\ &\quad c_i, d_i \geq 0, i = 0, 1, \ldots, n \end{aligned} \tag{11}$$

where ξ is an extremely small positive number and makes the influence of the term $\xi \sum_{i=0}^{n}(a_i^2 + c_i^2)$ in the objective function negligible. The constraints of the inclusion relations are equivalent to

$$Y_*(\boldsymbol{x}_j) \subseteq Y_j \Leftrightarrow$$
$$\begin{cases} y_j - e_j \leq (a_0 + \sum_{i=1}^{n} a_i x_{ij}) - (c_0 + \sum_{i=1}^{n} c_i |x_{ij}|) \\ (a_0 + \sum_{i=1}^{n} a_i x_{ij}) + (c_0 + \sum_{i=1}^{n} c_i |x_{ij}|) \leq y_j + e_j \end{cases} \tag{12}$$
$$Y_j \subseteq Y^*(\boldsymbol{x}_j) \Leftrightarrow$$
$$\begin{cases} (a_0 + \sum_{i=1}^{n} a_i x_{ij}) - (c_0 + \sum_{i=1}^{n} c_i |x_{ij}|) - (d_0 + \sum_{i=1}^{n} d_i |x_{ij}|) \leq y_j - e_j \\ y_j + e_j \leq (a_0 + \sum_{i=1}^{n} a_i x_{ij}) + (c_0 + \sum_{i=1}^{n} c_i |x_{ij}|) + (d_0 + \sum_{i=1}^{n} d_i |x_{ij}|) \end{cases} \tag{13}$$

3 Reduced Support Vector Machine

We now briefly introduce the basis of the theory of the reduced support vector machine (RSVM) [8].

Suppose that m training data $\{x_i, y_i\}$, $i = 1, 2, \ldots, m$ are given, where $x_i \in R^n$ are the input patterns and $y_i \in \{-1, 1\}$ are the related target values of two-class pattern classification case. Then the standard support vector machine with a linear kernel [16] is

$$\begin{aligned} &\min_{w,b,\xi} \ \frac{1}{2}\|w\|^2 + C(\sum_{i=1}^{m} \xi_i^2) \\ &s.t. \\ &\qquad y_i(w^t x_i + b) \geq 1 - \xi_i \\ &\qquad \xi_i \geq 0, i = 1, 2, \ldots, m \end{aligned} \tag{14}$$

where b is the location of hyperplane relative to the origin. The regularization constant $C > 0$ is the penalty parameter of the error term to determine the tradeoff between the flatness of linear functions $(w^t x_i + b)$ and empirical error.

In Lee and Mangasarian's approach [8], $b^2/2$ is added to the objective function of (14). This is equivalent to adding a constant feature to the training data and finding a separating hyperplane through the origin.

$$\begin{aligned} &\min_{w,b,\xi} \ \frac{1}{2}(\|w\|^2 + b^2) + C(\sum_{i=1}^{m} \xi_i^2) \\ &s.t. \\ &\qquad y_i(w^t x_i + b) \geq 1 - \xi_i \\ &\qquad \xi_i \geq 0, i = 1, 2, \ldots, m \end{aligned} \tag{15}$$

Its dual becomes the following bound-constrained problem

$$\begin{aligned} &\min_{\alpha} \ \frac{1}{2}\alpha^t(Q + \frac{I}{2C} + y^t y)\alpha - e^t\alpha \\ &s.t. \\ &\qquad 0 \leq \alpha_i, i = 1, 2, \ldots, m \end{aligned} \tag{16}$$

where e is the vector of all ones. Q is an $m \times m$ positive semi-definite matrix, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$ and $K(x_i, x_j) \equiv \phi(x_i)^t \phi(x_j)$ is the kernel function. In this case, $K(x_i, x_j) \equiv x_i^t x_j$.

In the optimal solution, w is a linear combination of training data

$$w = \sum_{i=1}^{m} y_i \alpha_i x_i \tag{17}$$

Substituting w into (15) by

$$y_i w^t x_i = \sum_{j=1}^{m} y_i y_j \alpha_j x_j^t x_i = \sum_{j=1}^{m} Q_{ij}\alpha_j = (Q\alpha)_i \tag{18}$$

$$\|w\|^2 = \sum_{i=1}^{m} y_i \alpha_i x_i^t w = \sum_{j=1}^{m} \alpha_i (Q\alpha)_i = \alpha^t Q \alpha \tag{19}$$

Then the problem becomes

$$\begin{aligned} &\min_{\alpha, b, \xi} \ \frac{1}{2}(\alpha^t Q \alpha + b^2) + C(\sum_{i=1}^{m} \xi_i^2) \\ &s.t. \\ &\qquad Q\alpha + by \geq e - \xi \end{aligned} \tag{20}$$

Dealing with large-scale data sets, the main idea of RSVM is to reduce the number of support vectors by randomly selecting a subset of k samples for w

$$w = \sum_{i \in K} y_i \alpha_i x_i \tag{21}$$

where K contains indices of the subset of k samples. The problem becomes (22) by substituting (21) with the number of major variables reduced to k

$$\begin{aligned} &\min_{\bar{\alpha}, b, \xi} \ \frac{1}{2}(\bar{\alpha}^t Q_{K,K} \bar{\alpha} + b^2) + C(\sum_{i=1}^{m} \xi_i^2) \\ &s.t. \\ &\qquad Q_{\cdot,K}\bar{\alpha} + by \geq e - \xi \end{aligned} \tag{22}$$

where $\bar{\alpha}$ is the collection of all α_i, $i \in K$. $Q_{\cdot,K}$ represents the sub-matrix of columns corresponding to K.

To simplify the term $1/2(\bar{\alpha}^t Q_{K,K}\bar{\alpha})$ to $1/2(\bar{\alpha}^t\bar{\alpha})$ following the generalized support vector machine (GSVM) by Mangasarian [9], we obtain the RSVM as follows

$$\begin{aligned} &\min_{\bar{\alpha},b,\xi}\ \frac{1}{2}(\bar{\alpha}^t\bar{\alpha}+b^2)+C(\sum_{i=1}^{m}\xi_i^2) \\ &s.t. \\ &\qquad Q_{\cdot,K}\bar{\alpha}+by \geq e-\xi \end{aligned} \tag{23}$$

For specific data sets, an appropriate nonlinear mapping $x \mapsto \phi(x)$ can be used to embed the original R^n features into a Hilbert feature space $\mathcal{F}$, $\phi : R^n \mapsto \mathcal{F}$, with a nonlinear kernel $K(x_i, x_j) \equiv \phi(x_i)^t\phi(x_j)$. The followings are well-known nonlinear kernels for regression problems, where γ, r, h, and θ are kernel parameters.

(1) $(\gamma x_i x_j^t + r)^h$: Polynomial kernel, $h \in \mathbb{N}$, $\gamma > 0$ and $r \geq 0$. [16]
(2) $e^{-\gamma\|x_i - x_j\|^2}$: Gaussian (radial basis) kernel, $\gamma > 0$. [10]
(3) $\tanh(\gamma x_i x_j^t + \theta)$: Hyperbolic tangent kernel, $\gamma > 0$. [11]

4 Proposed Methods

In this section we first propose the soft margin method to modify the excursion of separation margin and to be effective in the gray zone. Then the principle of RSVM is introduced to evaluate interval regression analysis.

4.1 Soft Margin Method

In a conventional SVM, the sign function is used as the decision-making function. The separation threshold of the sign function is 0. It results in an excursion of separation margin for unbalanced data sets. It is also a hard separation margin to be difficult to deal with the gray zone between classes. Thus the soft margin method is proposed to modify the excursion of separation margin and to be effective in the gray zone.

The soft margin is defined as

$$f^-(\varrho) = \frac{\arctan(-\varrho \cdot s + \vartheta \cdot s)}{\pi} + 0.5 \tag{24}$$

$$f^+(\varrho) = \frac{\arctan(\varrho \cdot s - \vartheta \cdot s)}{\pi} + 0.5 \tag{25}$$

where ϱ is the decision value. ϑ and s are offset parameter and scale parameter need to be estimated using statistical method.

With the soft margin, the predication of the class labels can be determined as follows

$$y(x) = \begin{cases} -1, \text{if } (\nu_r < f^-(\varrho) \text{ and } \varrho < \vartheta) \text{ or } (\nu_r > f^+(\varrho) \text{ and } \varrho > \vartheta) \\ +1, \text{if } (\nu_r > f^-(\varrho) \text{ and } \varrho < \vartheta) \text{ or } (\nu_r < f^+(\varrho) \text{ and } \varrho > \vartheta) \end{cases} \tag{26}$$

where ν_r is a random number between 0 and 1.

4.2 Interval Regression with RSVM

With the principle of RSVM, we can formulate the interval linear regression model as the following quadratic problem

$$\begin{aligned}
&\min_{\bar{a},\bar{c},\bar{d}} \ \frac{1}{2}(\bar{a}^t\bar{a}+\bar{c}^t\bar{c}+\bar{d}^t\bar{d}+b^2)+C(\sum_{j=1}^{q}\xi_j^2)\\
&s.t.\\
&\qquad Q_{\cdot,K}\bar{a}+by \geq e-\xi\\
&\qquad \bar{a}\boldsymbol{x}_j-\bar{c}|\boldsymbol{x}_j| \geq y_j-e_j\\
&\qquad \bar{a}\boldsymbol{x}_j+\bar{c}|\boldsymbol{x}_j| \leq y_j+e_j\\
&\qquad \bar{a}\boldsymbol{x}_j-\bar{c}|\boldsymbol{x}_j|-\bar{d}|\boldsymbol{x}_j| \leq y_j-e_j\\
&\qquad \bar{a}\boldsymbol{x}_j+\bar{c}|\boldsymbol{x}_j|+\bar{d}|\boldsymbol{x}_j| \geq y_j+e_j\\
&\qquad j=1,2,\ldots,q
\end{aligned} \tag{27}$$

where K contains indices of the subset by randomly selecting k samples. $\bar{a}$, $\bar{c}$, and $\bar{d}$ are the collections of all a_i, c_i, and d_i, $i \in K$, respectively. $Q_{\cdot,K}$ represents the sub-matrix of columns corresponding to K.

Given (27), the corresponding Lagrangian objective function is

$$\begin{aligned}
L := \ &\frac{1}{2}(\bar{a}^t\bar{a}+\bar{c}^t\bar{c}+\bar{d}^t\bar{d}+b^2)+C(\sum_{j=1}^{q}\xi_j^2)-\sum_{j=1}^{q}\alpha_{1j}(Q_{\cdot,K}\bar{a}+by_j-e+\xi_j)\\
&-\sum_{j=1}^{q}\alpha_{2j}(\bar{a}\boldsymbol{x}_j-\bar{c}|\boldsymbol{x}_j|-y_j+e_j)-\sum_{j=1}^{q}\alpha_{2j}^*(y_j+e_j-\bar{a}\boldsymbol{x}_j-\bar{c}|\boldsymbol{x}_j|)\\
&-\sum_{j=1}^{q}\alpha_{3j}(y_j-e_j-\bar{a}\boldsymbol{x}_j+\bar{c}|\boldsymbol{x}_j|+\bar{d}|\boldsymbol{x}_j|)\\
&-\sum_{j=1}^{q}\alpha_{3j}^*(\bar{a}\boldsymbol{x}_j+\bar{c}|\boldsymbol{x}_j|+\bar{d}|\boldsymbol{x}_j|-y_j-e_j)
\end{aligned} \tag{28}$$

Here L is Lagrangian and α_{1j}, α_{2j}, α_{2j}^*, α_{3j}, α_{3j}^* are Lagrange multipliers. The idea to construct a Lagrange function from the objective function and the corresponding constraints is to introduce a dual set of variables. It can be shown that the Lagrangian function has a saddle point with respect to the primal and dual variables in the solution.

The Karush-Kuhn-Tucker (KKT) conditions that the partial derivatives of L with respect to the primal variables $(\bar{a},\bar{c},\bar{d},\xi_j,b)$ for optimality

$$\frac{\partial L}{\partial \bar{a}}=0 \Rightarrow \bar{a}=\sum_{j=1}^{q}\alpha_{1j}Q_{\cdot,K}+\sum_{j=1}^{q}(\alpha_{2j}-\alpha_{2j}^*)\boldsymbol{x}_j-\sum_{j=1}^{q}(\alpha_{3j}-\alpha_{3j}^*)\boldsymbol{x}_j \tag{29}$$

$$\frac{\partial L}{\partial \bar{c}}=0 \Rightarrow \bar{c}=-\sum_{j=1}^{q}(\alpha_{2j}+\alpha_{2j}^*)|\boldsymbol{x}_j|+\sum_{j=1}^{q}(\alpha_{3j}+\alpha_{3j}^*)|\boldsymbol{x}_j| \tag{30}$$

$$\frac{\partial L}{\partial \bar{d}} = 0 \Rightarrow \bar{d} = \sum_{j=1}^{q} (\alpha_{3j} + \alpha^*_{3j})|\boldsymbol{x}_j| \tag{31}$$

$$\frac{\partial L}{\xi_j} = 0 \Rightarrow \xi_j = \frac{\alpha_{1j}}{2C} \tag{32}$$

$$\frac{\partial L}{b} = 0 \Rightarrow b = \sum_{j=1}^{q} \alpha_{1j} y_j \tag{33}$$

Substituting (29)–(33) in (28) gives the dual optimization problem

$$\begin{aligned}
\max\ & -\frac{1}{2}\Big(\sum_{i,j=1}^{q} \alpha_{1i}\alpha_{1j} Q_{\cdot,K}{}^{t} Q_{\cdot,K} + \sum_{i,j=1}^{q} (\alpha_{2i} - \alpha^*_{2i})(\alpha_{2j} - \alpha^*_{2j}) \boldsymbol{x}_i^t \boldsymbol{x}_j \\
& + \sum_{i,j=1}^{q} (\alpha_{3i} - \alpha^*_{3i})(\alpha_{3j} - \alpha^*_{3j}) \boldsymbol{x}_i^t \boldsymbol{x}_j + 2\sum_{i,j=1}^{q} \alpha_{1i} Q_{\cdot,K} (\alpha_{2j} - \alpha^*_{2j}) \boldsymbol{x}_j \\
& -2\sum_{i,j=1}^{q} \alpha_{1i} Q_{\cdot,K} (\alpha_{3j} - \alpha^*_{3j}) \boldsymbol{x}_j - 2\sum_{i,j=1}^{q} (\alpha_{2i} - \alpha^*_{2i})(\alpha_{3j} - \alpha^*_{3j}) \boldsymbol{x}_i^t \boldsymbol{x}_j \\
& + \sum_{i,j=1}^{q} (\alpha_{2i} + \alpha^*_{2i})(\alpha_{2j} + \alpha^*_{2j}) |\boldsymbol{x}_i|^t |\boldsymbol{x}_j| \\
& -2\sum_{i,j=1}^{q} (\alpha_{2i} + \alpha^*_{2i})(\alpha_{3j} + \alpha^*_{3j}) |\boldsymbol{x}_i|^t |\boldsymbol{x}_j| \\
& +2\sum_{i,j=1}^{q} (\alpha_{3i} + \alpha^*_{3i})(\alpha_{3j} + \alpha^*_{3j}) |\boldsymbol{x}_i|^t |\boldsymbol{x}_j|\Big) \\
& -\frac{1}{4C}\sum_{j=1}^{q} \alpha_{1j}^2 + \sum_{j=1}^{q} (\alpha_{2j} - \alpha^*_{2j}) y_j - \sum_{j=1}^{q} (\alpha_{2j} + \alpha^*_{2j}) e_j \\
& -\sum_{j=1}^{q} (\alpha_{3j} - \alpha^*_{3j}) y_j + \sum_{j=1}^{q} (\alpha_{3j} + \alpha^*_{3j}) e_j \\
s.t.\ & \\
& \alpha_{1j}, \alpha_{2j}, \alpha^*_{2j}, \alpha_{3j}, \alpha^*_{3j} \geq 0
\end{aligned} \tag{34}$$

Similarly, we can obtain the interval nonlinear regression model by mapping $x \mapsto \phi(x)$ to embed the original R^n features into a Hilbert feature space $\mathcal{F}$, $\phi : R^n \mapsto \mathcal{F}$, with a nonlinear kernel $K(x_i, x_j) \equiv \phi(x_i)^t \phi(x_j)$ as discussed in Section 3. Therefore, by replacing $\boldsymbol{x}_i^t \boldsymbol{x}_j$ and $|\boldsymbol{x}_i|^t |\boldsymbol{x}_j|$ in (34) with $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $K(|\boldsymbol{x}_i|, |\boldsymbol{x}_j|)$, respectively, we obtain the dual optimization problem as (35)

$$
\begin{aligned}
\max\ & -\frac{1}{2}(\sum_{i,j=1}^{q} \alpha_{1i}\alpha_{1j}Q_{\cdot,K}{}^{t}Q_{\cdot,K} + \sum_{i,j=1}^{q} (\alpha_{2i}-\alpha_{2i}^{*})(\alpha_{2j}-\alpha_{2j}^{*})K(\boldsymbol{x}_i,\boldsymbol{x}_j) \\
& + \sum_{i,j=1}^{q} (\alpha_{3i}-\alpha_{3i}^{*})(\alpha_{3j}-\alpha_{3j}^{*})K(\boldsymbol{x}_i,\boldsymbol{x}_j) \\
& +2 \sum_{i,j=1}^{q} \alpha_{1i}Q_{\cdot,K}(\alpha_{2j}-\alpha_{2j}^{*})\boldsymbol{x}_j - 2\textstyle\sum_{i,j=1}^{q} \alpha_{1i}Q_{\cdot,K}(\alpha_{3j}-\alpha_{3j}^{*})\boldsymbol{x}_j \\
& -2 \sum_{i,j=1}^{q} (\alpha_{2i}-\alpha_{2i}^{*})(\alpha_{3j}-\alpha_{3j}^{*})K(\boldsymbol{x}_i,\boldsymbol{x}_j) \\
& + \sum_{i,j=1}^{q} (\alpha_{2i}+\alpha_{2i}^{*})(\alpha_{2j}+\alpha_{2j}^{*})K(|\boldsymbol{x}_i|,|\boldsymbol{x}_j|) \\
& -2 \sum_{i,j=1}^{q} (\alpha_{2i}+\alpha_{2i}^{*})(\alpha_{3j}+\alpha_{3j}^{*})K(|\boldsymbol{x}_i|,|\boldsymbol{x}_j|) \\
& +2 \sum_{i,j=1}^{q} (\alpha_{3i}+\alpha_{3i}^{*})(\alpha_{3j}+\alpha_{3j}^{*})K(|\boldsymbol{x}_i|,|\boldsymbol{x}_j|)) \\
& -\frac{1}{4C}\sum_{j=1}^{q}\alpha_{1j}^{2} + \sum_{j=1}^{q}(\alpha_{2j}-\alpha_{2j}^{*})y_j - \sum_{j=1}^{q}(\alpha_{2j}+\alpha_{2j}^{*})e_j \\
& -\sum_{j=1}^{q}(\alpha_{3j}-\alpha_{3j}^{*})y_j + \sum_{j=1}^{q}(\alpha_{3j}+\alpha_{3j}^{*})e_j \\
s.t.\ & \\
& \alpha_{1j},\alpha_{2j},\alpha_{2j}^{*},\alpha_{3j},\alpha_{3j}^{*} \geq 0 \qquad (35)
\end{aligned}
$$

5 Conclusion

This paper proposes a RSVM approach in evaluating interval regression models. The main idea of RSVM is to reduce the number of support vectors by randomly selecting a subset of samples. In addition the soft margin method is proposed to modify the excursion of separation margin and to be effective in the gray zone.

In this paper we estimate the interval regression model with crisp inputs and interval output. In future works, both interval inputs–interval output and fuzzy inputs–fuzzy output will be considered.

Acknowledgement

The authors appreciate the anonymous referees for their careful reading and the fruitful comments for the manuscript. Also special thanks to National Science Council in Taiwan for supporting this study (C.H. Huang: NSC 97-2410-H-424-019; H.Y. Kao: NSC 97-2410-H-259-065).

References

1. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2, 121–167 (1998)
2. Cortes, C., Vapnik, V.N.: Support Vector Networks. Machine Learning 20, 273–297 (1995)
3. Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A., Vapnik, V.N.: Support Vector Regression Machines. In: Mozer, M., Jordan, M., Petsche, T. (eds.) Advances in Neural Information Processing Systems, vol. 9, pp. 155–161. MIT Press, Cambridge (1997)
4. Hong, D.H., Hwang, C.H.: Support Vector Fuzzy Regression Machines. Fuzzy Sets and Systems 138, 271–281 (2003)
5. Hong, D.H., Hwang, C.H.: Interval Regression Analysis using Quadratic Loss Support Vector Machine. IEEE Transactions on Fuzzy Systems 13, 229–237 (2005)
6. Jeng, J.T., Chuang, C.C., Su, S.F.: Support Vector Interval Regression Networks for Interval Regression Analysis. Fuzzy Sets and Systems 138, 283–300 (2003)
7. Kacprzyk, J., Fedrizzi, M.: Fuzzy Regression Analysis. Physica-Verlag, Heidelberg (1992)
8. Lee, Y.J., Mangasarian, O.L.: RSVM: Reduced Support Vector Machines. In: Proceedings of 1st SIAM International Conference on Data Mining (2001)
9. Mangasarian, O.L.: Generalized Support Vector Machines. In: Smola, A.J., Bartlett, P.L., Schölkopf, B., Schuurmans, D. (eds.) Advances in Large Margin Classifiers, pp. 135–146. MIT Press, Cambridge (2000)
10. Micchelli, C.A.: Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions. Constructive Approximation 2, 11–22 (1986)
11. Schölkopf, B., Burges, C.J.C., Smola, A.J. (eds.): Advances in Kernel Methods: Support Vector Learning. MIT Press, Cambridge (1999)
12. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2002)
13. Smola, A.J., Schölkopf, B.: A Tutorial on Support Vector Regression. NeuroCOLT2 Tech. Report, NeuroCOLT (1998); Statistics and Computing 14, 199–222 (2004)
14. Tanaka, H., Lee, H.: Interval Regression Analysis by Quadratic Programming Approach. IEEE Transactions on Fuzzy Systems 6, 473–481 (1998)
15. Tanaka, H., Uejima, S., Asai, K.: Fuzzy Linear Regression Model. IEEE Transactions on Systems, Man and Cybernetics 10, 2933–2938 (1980)
16. Vapnik, V.N.: Statistical Learning Theory. John Wiley and Sons, Inc., New York (1998)

Author Index