Jiuyong Li (Ed.)

LNAI 6464

AI 2010: Advances in Artificial Intelligence

23rd Australasian Joint Conference
Adelaide, Australia, December 2010
Proceedings

Lecture Notes in Artificial Intelligence 6464

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

Subseries of Lecture Notes in Computer Science

Jiuyong Li (Ed.)

AI 2010: Advances in Artificial Intelligence

23rd Australasian Joint Conference
Adelaide, Australia, December 2010
Proceedings

Springer

Volume Editor

Jiuyong Li
School of Computer and Information Science
University of South Australia
Mawson Lakes, SA, 5095, Australia
E-mail: jiuyong.li@unisa.edu.au

Library of Congress Control Number: 2010939806

CR Subject Classification (1998): I.2, H.3, H.4, F.1, H.2.8, I.5

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-642-17431-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-17431-5 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

This volume contains papers presented at AI 2010, the 23rd Australasian Joint Conference on Artificial Intelligence held December 7–10, 2010 in Adelaide, South Australia. The Australasian Joint Conference on Artificial Intelligence has a long history and has established itself as the premier event for artificial intelligence researchers in the Australasia region.

AI 2010 received 112 submissions with authors from 27 countries. The countries most represented included Australia, New Zealand, China, Malaysia, Japan, Iran, Canada, and Spain. Each submission was reviewed by up to 6 reviewers, and on average there were 3.2 reviewers per paper. Some papers were intensively discussed among the reviewers, and extra reviews were sought during the discussion period. After the intensive review process, the Program Committee decided to accept 52 papers for presentation at the conference. These papers are included in this volume. The research papers cover a range of topics in artificial intelligence theory and application. A strong focus in this volume is on various learning methods and their applications.

AI 2010 featured three keynote speeches by distinguished researchers: Gautam Biswas (Vanderbilt University, USA), Kotagiri Ramamohanarao (The University of Melbourne, Australia), and Qiang Yang (The Hong Kong University of Science and Technology). Their talks cover theoretical challenges and advanced applications in current artificial intelligence research.

Two workshops, with their own proceedings, were held on December 7. Extreme Learning Machines (ELM 2010) was organized by Guang-Bin Huang (Nanyang Technological University, Singapore) and Dianhui Wang (La Trobe University, Australia). The 6th Australasian Ontology Workshop was organized by Thomas Meyer (Meraka Institute South Africa), Mehmet Orgun (Macquarie University, Australia) and Kerry Taylor (CSIRO ICT Centre, Australia).

AI 2010 would not be successful without the support of authors, reviewers and organizers. We thank the many authors for submitting their research papers to AI 2010. We thank the successful authors whose papers are published in this volume for their collaboration during the preparation of final submissions. We appreciate all Program Committee members, including external reviewers, for their timely reviews working to a tight schedule. We are grateful to the senior Program Committee members for Organizing discussions and shortlisting papers. We also thank members of the Organizing Committee for their year-long efforts in the preparation, promotion and organization of the conference, especially Ivan Lee for his outstanding service to the conference. We thank support provided by the Australian Computer Society National Committee for Artificial Intelligence.

The conference was held at the City West Campus of the University of South Australia (UniSA). We thank UniSA for sponsoring the venues and facilities.

We acknowledge the assistance provided by EasyChair for conference management, which saved us a lot of time and effort. We appreciate the professional service provided by the Springer LNCS editorial and publishing teams.

September 2010 Jiuyong Li

Conference Organization

Conference Chairs

Markus Stumptner	University of South Australia, Australia
Graham Williams	Australian Taxation Office, Australia

Program Chairs

Jiuyong Li	University of South Australia, Australia
John Debenham	University of Technology, Sydney, Australia

Organizing Chairs

Ivan Lee	University of South Australia, Australia

Workshop Chair

Dianhui Wang	La Trobe University, Australia

Publicity Chair

Junbin Gao	Charles Sturt University, Australia

Publication Chair

Mengjie Zhang	Victoria University of Wellington, New Zealand

Senior Program Committee

James Bailey	The University of Melbourne, Australia
Dan Corbett	DARPA, Washington, D.C., USA
Stephen Cranefield	University of Otago, New Zealand
David Dowe	Monash University, Australia
Byeong Ho Kang	University of Tasmania, Australia
Reinhard Klette	The University of Auckland, New Zealand
Zbigniew Michalewicz	The University of Adelaide, Australia
Mehmet Orgun	Macquarie University, Australia

Dianhui Wang	La Trobe University, Australia
Wayne Wobcke	The University of New South Wales, Australia
Chengqi Zhang	University of Technology, Sydney, Australia
Mengjie Zhang	Victoria University of Wellington, New Zealand
Yan Zhang	University of Western Sydney, Australia
Zhi-Hua Zhou	Nanjing University, China

Program Committee

Hussein Abbass	The University of New South Wales at ADFA, Australia
David Albrecht	Monash University, Australia
Peter Andreae	Victoria University of Wellington, New Zealand
Yun Bai	University of Western Sydney, Australia
Mike Bain	The University of New South Wales, Australia
Tim Baldwin	The University of Melbourne, Australia
Nick Bassiliades	Aristotle University of Thessaloniki, Greece
Peter Baumgartner	NICTA and The Australian National University, Australia
Mikael Boden	The University of Queensland, Australia
Richard Booth	University of Luxembourg and Mahasarakham University, Thailand
Sebastian Brand	NICTA, Australia
Lam Thu Bui	Le Quy Don University, Vietnam
Wray Buntine	NICTA, Australia
Jinhai Cai	Queensland University of Technology, Australia
Longbing Cao	University of Technology, Sydney, Australia
Lawrence Cavedon	NICTA, Australia
Chia-Yen Chen	National University of Kaohsiung, Taiwan
Hui Chen	Shandong University, China
Ling Chen	University of Technology, Sydney, Australia
Songcan Chen	Nanjing University of Aeronautics and Astronautics, China
Sung-Bae Cho	Yonsei University, Korea
Grace Chung	Google, Australia
Honghua Dai	Deakin University, Australia
Hepu Deng	RMIT University, Australia
Jeremiah Deng	University of Otago, New Zealand
Joachim Denzler	Friedrich Schiller University of Jena, Germany
Grant Dick	University of Otago, New Zealand
Yulin Ding	Defence Science and Technology Organisation, Australia
Trevor Dix	Monash University, Australia
Roland Dodd	CQUniversity, Australia

Xiangjun Dong	Shandong Institute of Light Industry, China
Atilla Elçi	Middle East Technical University Northern Cyprus Campus
Esra Erdem	Sabancı University, Turkey
Daryl Essam	The University of New South Wales at ADFA, Australia
Marcus Frean	Victoria University of Wellington, New Zealand
Alfredo Gabaldon	New University of Lisbon, Portugal
Marcus R Gallagher	The University of Queensland, Australia
Xiaoying Gao	Victoria University of Wellington, New Zealand
Yang Gao	Nanjing University, China
Manolis Gergatsoulis	Ionian University, Greece
Guido Governatori	NICTA, Australia
Hans W. Guesgen	Massey University, New Zealand
Fikret Gurgen	Bogazici University, Turkey
Bernhard Hengst	The University of New South Wales, Australia
Jose Hernandez-Orallo	Polytechnic University of Valencia, Spain
Geoffrey Holmes	University of Waikato, New Zealand
Wei-Chiang Hong	Zhejiang University of Finance and Economics, China
Xiaodi Huang	Charles Sturt University, Australia
Marcus Hutter	The Australian National University, Australia
Lakhmi Jain	University of South Australia, Australia
Huidong (Warren) Jin	CSIRO MIS, Australia
Zhi Jin	Institute of Mathematics, Chinese Academy of Sciences, China
Ken Kaneiwa	National Institute of Information and Communications Technology, Japan
George Katsirelos	The University of Lille-Nord de France, France
Paul Kennedy	University of Technology, Sydney, Australia
Philip Kilby	NICTA and The Australian National University, Australia
Frank Klawonn	Ostfalia University of Applied Sciences, Germany
Alistair Knott	University of Otago, New Zealand
Mario Koeppen	Kyushu Institute of Technology, Japan
Kevin Korb	Monash University, Australia
Norbert Krüger	The Maersk Mc-Kinney Moller Institute for Production Technology, Denmark
Rex Kwok	The University of New South Wales, Australia
Gerhard Lakemeyer	RWTH Aachen University, Germany
Jérôme Lang	LAMSADE, France
Maria R. Lee	Shih Chien University, Taiwan

Bin Li	University of Science and Technology of China, China
Gang Li	Deakin University, Australia
Li Li	Southwest University, China
Ming Li	Nanjing University, China
Wei Li	CQUniversity Australia, Australia
Yuefeng Li	Queensland University of Technology, Australia
Feng Ling	Tsinghua University, China
Jing Liu	Xidian University, China
Lin Liu	University of South Australia, Australia
Wanquan Liu	Curtin University of Technology, Australia
Abdun Naser Mahmood	RMIT University, Australia
Eric Martin	The University of New South Wales, Australia
Brendan McCane	University of Otago, New Zealand
Kathryn Merrick	The University of New South Wales at ADFA, Australia
Thomas Meyer	Meraka Institute, South Africa
Diego Molla-Aliod	Macquarie University, Australia
John Morris	The University of Auckland, New Zealand
Saeid Nahavandi	Deakin University, Australia
Detlef Nauck	BT, UK
Richi Nayak	Queensland University of Technology, Australia
Kourosh Neshatian	Victoria University of Wellington, New Zealand
David Newth	CSIRO, Australia
Philip Nguyen	Government of South Australia, Australia
Kouzou Ohara	Aoyama-Gakuin University, Japan
Maurice Pagnucco	The University of New South Wales, Australia
Linqiang Pan	Huazhong University of Science and Technology, China
Francis Jeffry Pelletier	Simon Fraser University, Canada
Wei Peng	RMIT University, Australia
Duc Nghia Pham	NICTA, Australia
David Powers	Flinders University, Australia
Mikhail Prokopenko	CSIRO ICT Centre, Australia
Cèsar Ferri Ramírez	Polytechnic University of Valencia, Spain
Tapabrata Ray	The University of New South Wales at ADFA, Australia
Ralf Reulke	Humboldt University of Berlin, Germany
Jeff Riley	RMIT University, Australia
Panos Rondogiannis	University of Athens, Greece
Suzanne Sadedin	Monash University, Australia
Ruhul Sarker	The University of New South Wales at ADFA, Australia

Torsten Schaub	University of Potsdam, Germany
Rolf Schwitter	Macquarie University, Australia
Tony Smith	Waikato University, New Zealand
Bing-Yu Sun	Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, China
Xiaoxun Sun	Australian Council for Educational Research, Australia
Maolin Tang	Queensland University of Technology, Australia
Mutsuhiro Terauchi	Hiroshima International University, Japan
Michael Thielscher	The University of New South Wales, Australia
Simon Thompson	BT, UK
Peter Tischer	Monash University, Australia
Andrea Torsello	Ca′ Foscari University of Venice, Italy
Miroslav Velev	Aries Design Automation, USA
Toby Walsh	NICTA and The University of New South Wales, Australia
Kewen Wang	Griffith University, Australia
Peter Whigham	University of Otago, New Zealand
William Wilson	The University of New South Wales, Australia
Brendon J. Woodford	University of Otago, New Zealand
Jianxin Wu	Nanyang Technological University, Singapore
Roland Yap	National University of Singapore, Singapore
Jian Yu	Beijing Jiaotong University, China
Daoqiang Zhang	Nanjing University of Aeronautics and Astronautics, China
Dongmo Zhang	University of Western Sydney, Australia
Haolan Zhang	RMIT University, Australia
Ji Zhang	University of Southern Queensland, Australia
Jun Zhang	Sun Yat-Sen University, China
Min-Ling Zhang	Hohai University, China
Shichao Zhang	University of Technology, Sydney, Australia
Xiuzhen Zhang	RMIT University, Australia
Yanchang Zhao	Centrelink, Australia
Yi Zhou	University of Western Sydney, Australia
Xingquan (Hill) Zhu	University of Technology, Sydney, Australia

External Reviewers

Antonio Bella	Daniel Harabor
Arina Britz	Sumit Jha
Weiling Cai	Samantha Kleinberg
Angelos Charalambidis	Vassilios Kountouriotis
Jinyong Chen	Kalliopi Kravari
Ni Chen	Jason Jingshi Li
Wei-Neng Chen	Jinjiu Li

Yuan-long Li
Bo Liu
Chao Luo
Wei Luo
Georgios Meditskos
Alexander Mitev
Nina Narodytska
Hongjing Peng
Gavin Rens
Pnina Soffer
Marius Schneider
Chew Lim Tan
Sven Thiele
William Uther
Ivan Varzinczak
Can Wang
Zhe Wang
Wei-jie Yu
Zhi-hui Zhan
Xian Zhang
Dengji Zhao
Jinghui Zhong

Other Organization Members

Webmaster: Zhenglin Wang (University of South Australia)
Assistant to PC Chairs: Muzammil M. Baig (University of South Australia)

Table of Contents

Knowledge Representation and Reasoning

Data Mining and Knowledge Discovery

Machine Learning

Statistical Learning

Evolutionary Computation

Particle Swarm Optimization

Intelligent Agent

Search and Planning

Natural Language Processing

AI Applications

Pushing the Envelope: General Game Players Prove Theorems

Sebastian Haufe [1] and Michael Thielscher [2]

[1] Department of Computer Science
Dresden University of Technology
sebastian.haufe@mailbox.tu-dresden.de
[2] School of Computer Science and Engineering
The University of New South Wales
mit@cse.unsw.edu.au

Abstract. A *general* game player is a system that can play previously unknown games given nothing but their rules. A key to success in this endeavour is the ability to automatically gain knowledge about new games that follows from the rules without being explicitly given. In this paper, we show how a recently developed, theoretical method for automated theorem proving in general game playing can be put into practice. To this end, we extend the method so as to allow a general game player to systematically search and verify multiple temporal game properties at once. We formally prove this extension to be correct, and we report on extensive experiments that show how this improvement helps to significantly enhance the ability of a successful general game player to infer new properties about a previously unknown game.

1 Introduction

General game playing is concerned with the development of systems that understand the rules of previously unknown games and learn to play these games well without human intervention. Identified as a Grand Challenge for AI, this endeavour requires to combine methods from a variety of sub-disciplines, including automated reasoning, search, game playing, and learning [9,5,3,11,2].

A key capability is to automatically gain knowledge about games that follows from the rules without being explicitly given. In [12,13] we have laid the foundations for the use of Answer Set Programming [4] to automatically prove properties of a game from its mere rules. While initial experiments had shown that this provides a viable method for a general game player to establish the truth of a specific property, the practice of general game playing requires a player to systematically search large sets of potentially valid and useful properties in order to find those that actually hold [5,3,11]. Proving each candidate formula individually constitutes a considerable computational burden [13].

In this paper, we extend the method in [13] so as to allow a general game player to systematically search through and verify multiple temporal game properties at once. The correctness of our extended approach is formally proved, and we

J. Li (Ed.): AI 2010, LNAI 6464, pp. 1–10, 2010.

report on extensive experiments that show how our improvement significantly enhances the ability of a successful general game player to infer new properties about a previously unknown game.

2 Background

2.1 Game Description Language

The Game Description Language (GDL) has been developed to formalise the rules of any finite $n \geq 1$-player game with complete information in such a way that the description can be automatically processed by a general game player. Due to lack of space, we can give just a very brief introduction to GDL and have to refer to [7] for details.

GDL is based on the standard syntax of logic programs, including negation. We assume familiarity with the basic notions of logic programming. We adopt the Prolog convention according to which variables are denoted by uppercase letters and predicate and function symbols start with a lowercase letter. As a tailor-made specification language, GDL uses a few pre-defined predicate symbols:

`role(R)`	`R` is a player
`init(F)`	`F` holds in the initial position
`true(F)`	`F` holds in the current position
`legal(R,M)`	player `R` has legal move `M`
`does(R,M)`	player `R` does move `M`
`next(F)`	`F` holds in the next position
`terminal`	the current position is terminal
`goal(R,N)`	player `R` gets goal value `N`

A further standard predicate is `distinct(X,Y)`, which means syntactic inequality of the two arguments. GDL imposes restrictions on the use of these keywords:

- `role` only appears in facts (i.e., clauses with empty body);
- `init` and `next` only appear as head of clauses, and `init` does not depend on any of `true`, `legal`, `does`, `next`, `terminal`, or `goal`;
- `true` and `does` only appear in clause bodies with `does` not depending on any of `legal`, `terminal`, or `goal`.

Additional general restrictions are placed on a set of rules with the intention to ensure finiteness of the set of derivable predicate instances. Specifically, the set of rules must be *stratified* [1] and *allowed* [6]. Stratified logic programs are known to admit a unique *standard model* [1]. As an example, Figure 1 shows an excerpt of a GDL description for a game called "Quarto."

Based on the concept of the standard model, a GDL description can be understood as a state transition system as follows [7]. To begin with, any valid game description G in GDL contains a finite set of function symbols, including constants, which implicitly determines a set of ground terms Σ. This set constitutes the symbol base Σ in the formal semantics for G.

```
role(r1).  role(r2).    init(cell(1,1,b)).  ...  init(cell(4,4,b)).
init(sctrl(r1)).        init(pool(p0000)).  ...  init(pool(p1111)).

legal(R,select(P))    :- true(sctrl(R)),true(pool(P)).
legal(R,place(P,M,N)) :- true(pctrl(R)),true(slctd(P)),true(cell(M,N,b)).
legal(R,noop)         :- role(R),not true(sctrl(R)),not true(pctrl(R)).

next(pool(P))     :- true(pool(P)),not does(r1,select(P)),
                                   not does(r2,select(P)).
next(slctd(P))    :- does(R,select(P)).
next(cell(M,N,P)) :- does(R,place(P,M,N)).
next(cell(M,N,P)) :- true(cell(M,N,P)),does(R,select(P)).
next(cell(M,N,P)) :- true(cell(M,N,P)),does(R,place(P,S,T)),!=(M,N,S,T).

next(sctrl(R))  :- true(pctrl(R)).
next(pctrl(r1)) :- true(sctrl(r2)).
next(pctrl(r2)) :- true(sctrl(r1)).
```

Fig. 1. A GDL description of "Quarto" (without definitions for termination, goal values, and `!=/4`). Two players take turns selecting one of 16 jointly used 4-attributed pieces $p0000, p0001, \ldots, p1111$ which the other player must place on a 4×4 board. The player wins who completes a line of 4 pieces with a common attribute.

The players R and the initial position of a game can be directly determined from the clauses for `role` and `init`, respectively. In order to determine the legal moves, update, termination, and outcome (i.e., goal values) for a given position, this position has to be encoded first, using the keyword `true`. To this end, for any *finite* subset $S = \{f_1, \ldots, f_n\} \subseteq \Sigma$ of a set of ground terms, the following set of logic program facts encodes S as the current position:

$$S^{\texttt{true}} \stackrel{\text{def}}{=} \{\texttt{true}(f_1)., \ \ldots, \ \texttt{true}(f_n).\}$$

Furthermore, for any function $A : (\{r_1, \ldots, r_k\} \mapsto \Sigma)$ that assigns a move to each player $r_1, \ldots, r_k \in R$, the following set of facts encodes A as a joint move:

$$A^{\texttt{does}} \stackrel{\text{def}}{=} \{\texttt{does}(r_1, A(r_1))., \ \ldots, \ \texttt{does}(r_k, A(r_k)).\}$$

Definition 1. *Let G be a GDL specification whose signature determines ground terms Σ. The* semantics *of G is the state transition system $(R, S_{init}, T, l, u, g)$ where*[1]

- $R = \{r : G \models \texttt{role}(r)\}$ *(the players);*
- $S_{init} = \{f : G \models \texttt{init}(f)\}$ *(the initial position);*
- $T = \{S : G \cup S^{\texttt{true}} \models \texttt{terminal}\}$ *(the terminal positions);*
- $l = \{(r, a, S) : G \cup S^{\texttt{true}} \models \texttt{legal}(r, a)\}$ *(the legality relation);*

[1] Below, entailment $\models$ is via the aforementioned standard model for stratified clause sets.

– $u(A, S) = \{f : G \cup S^{\texttt{true}} \cup A^{\texttt{does}} \models \texttt{next}(f)\}$ *(the update function);*
– $g = \{(r, v, S) : G \cup S^{\texttt{true}} \models \texttt{goal}(r, v)\}$ *(the goal relation).*

We write $S \xrightarrow{A} S'$ *if* $A : (R \mapsto \Sigma)$ *is such that* $(r, A(r), S) \in l$ *for each* $r \in R$ *and* $S' = u(A, S)$ *(and* $S \notin T$*). We call* $S_0 \xrightarrow{A_0} S_1 \xrightarrow{A_1} \ldots \xrightarrow{A_{m-1}} S_m$ *(where* $m \geq 0$*) a* sequence *(of legal moves), sometimes abbreviated as* $(S_0, S_1, \ldots, S_m)$*. A state* S *is called* reachable *iff there is a sequence which starts in the initial state* S_{init} *and ends in* S*.*

This definition provides a formal semantics by which a GDL description is interpreted as an abstract k-player game: in every position S, starting with S_{init}, each player r chooses a move a that is legal, i.e., satisfies $l(r, a, S)$. As a consequence the game state changes to $u(A, S)$, where A is the joint move. The game ends if a position in T is reached, and then g determines the outcome. The restrictions in GDL ensure that entailment w.r.t. the standard model is decidable and that only finitely many instances of each predicate are entailed. This guarantees that the definition of the semantics is effective [7].

2.2 Formalising and Encoding Temporal Game Properties

Next, we briefly summarise syntax and semantics of a language for formulating individual game properties. We also recapitulate from [13] the so-called *temporal GDL extension*, which is needed for proving properties given in this language.

Definition 2. *The set of* formulas *is (1) based on all ground atoms over the signature of a GDL description which are different from* `init` *and* `next` *and not dependent on* `does`*, and (2) closed under* $\neg, \wedge, \vee, \supset, \bigcirc$*. The* degree *of a formula* φ *is the maximal "nesting" of the unary* $\bigcirc$*-operator in* φ*.*

Modality $\bigcirc\varphi$ states that φ holds in all positions that are a direct, legal successor of the current game state. An example property in the Quarto game is the periodic return of "select control" to player *r1* every four moves, which can be formulated via the formula $true(sctrl(r1)) \supset \bigcirc^4 true(sctrl(r1))$ with degree 4.

A formula with degree n follows from a GDL description if it holds w.r.t. all sequences of length n and all shorter sequences that end in a terminal state [13].

Definition 3. *A sequence is called* n-max *iff it is of length* n*, or shorter and ending in a terminal state. Let* G *be a GDL description and* φ *a formula with degree* n*. We say that* S_0 satisfies φ *(written* $S_0 \models_t \varphi$*) if for all* n*-max sequences* $(S_0, \ldots, S_m)$ $(m \leq n)$ *we have that* $(S_0, \ldots, S_m) \models_t \varphi$ *according to the following definition:*

$(S_i, \ldots, S_m) \models_t p$ *iff* $G \cup S_i^{\texttt{true}} \models p$ *(p ground atom)*
$(S_i, \ldots, S_m) \models_t \neg\varphi$ *iff* $(S_i, \ldots, S_m) \not\models_t \varphi$ *(likewise for* $\wedge, \vee, \supset$*)*
$(S_i, \ldots, S_m) \models_t \bigcirc\varphi$ *iff* $i = m$ *or* $(S_{i+1}, \ldots, S_m) \models_t \varphi$

Automatically verifying properties over sequences of successive game states against a given GDL specification G requires to build the *temporal extension*

of G (with some horizon n), denoted G_n. It is obtained by joining timed variants of G (which enrich predicates with a time argument) for each time level $0 \leq i \leq n$, omitting `does`-dependent rules for level n. We refer to [13] for a formal definition and just give an example: Consider the fourth rule with head `next` in the GDL description G of Figure 1. It depends on `does`, hence the following timed variant is contained in G_n for every $0 \leq i \leq n-1$:

$$\texttt{true(cell(M,N,P),}i+1\texttt{)} \texttt{ :- true(cell(M,N,P),}i\texttt{),}\ \texttt{does(R,select(P),}i\texttt{).}$$

The definitions of $S^{\texttt{true}}$ and $A^{\texttt{does}}$ (cf. Section 2.1) are similarly extended to $S^{\texttt{true}}(0)$ and $A^{\texttt{does}}(i)$, respectively. The encoding of a formula φ can now be related to a temporally extended GDL description G_n in a way that corresponds to formula entailment w.r.t. G.

Definition 4. *Let $\eta(\varphi)$ be a 0-ary atom which represents a unique name for formula φ with degree n. An* encoding of φ, *denoted $Enc(\varphi)$, is a stratified set of rules whose heads include $\eta(\varphi)$ and do not occur elsewhere, and such that for all n-max sequences $S_0 \overset{A_0}{\rightarrow} S_1 \ldots \overset{A_{m-1}}{\rightarrow} S_m$ of a GDL description G:*

$$(S_0, \ldots, S_m) \models_t \varphi \quad \textit{iff} \quad S_0^{\texttt{true}}(0) \cup G_n \cup \bigcup_{i=0}^{m-1} A_i^{\texttt{does}}(i) \cup Enc(\varphi) \models \eta(\varphi)$$

In the following we assume Enc to be given, whose construction can be easily automated. Recall, e.g., $\varphi = true(sctrl(r1)) \supset \bigcirc^4 true(sctrl(r1))$ from above and let $\eta(\varphi) = \texttt{a}$, then the following set of rules encode φ:

```
a :- not true(sctrl(r1),0).  a :- terminal(0).  a :- terminal(1).
a :-     true(sctrl(r1),4).  a :- terminal(2).  a :- terminal(3).
```

3 Proving Multiple Temporal Game Properties at Once

In [13] we have shown how the encoding of a game property (i.e., a temporal formula), together with the temporal extension of a given set of game rules, can be fed into a system for Answer Set Programming (ASP) in order to establish whether the rules entail the property.[2] Even though being the currently fastest approach for calculating models of logic programs, requiring a general game player to evoke an ASP system individually for each formula in a large set of candidate properties is not feasible for the practice of general game playing with a limited amount of time to analyse the rules of a hitherto unknown game.

In the following we therefore develop a crucial extension of our method that enables a general game player to evoke an ASP system only once in order to determine precisely which of a whole set Φ of formulas is valid w.r.t. a given game description. For this purpose, we construct two answer set programs for Φ, one to establish base case proofs and one for the induction steps. For any $\varphi \in \Phi$, then, if all answer sets for the base case program satisfy φ, then φ is

[2] *Answer sets* are specific models of logic programs with negation; see e.g. [4].

entailed in the initial state. If additionally all answer sets of the induction step program satisfy $\varphi \supset \bigcirc\varphi$, we can conclude that φ is entailed in all reachable states. The encoding of each player performing a legal move in each nonterminal state is given by a set of ASP clauses P_n^{legal}, consisting of a set of negation-free clauses which defines the domains of actions (`adom`) and the following clauses for each $0 \leq i \leq n$:[3]

(c_1) `terminated(`i`) :- terminal(`i`).`
(c_2) `terminated(`i`) :- terminated(`$i-1$`).`
(c_3) `1{does(R,A,`i`):adom(R,A)}1 :- role(R), not terminated(`i`).`
(c_4) `:- does(R,A,`i`), not legal(R,A,`i`).`

For a GDL description G and a finite set of formulas Φ with maximal degree $\widehat{n}$, the answer set program for the *base case* is defined as follows:

$$P_\Phi^{bc}(G) = S_{init}^{\texttt{true}}(0) \cup G_{\widehat{n}} \cup P_{\widehat{n}-1}^{legal} \cup \bigcup_{\varphi \in \Phi} Enc(\varphi)$$

Put in words, $P_\Phi^{bc}(G)$ consists of an encoding for the initial state, $S_{init}^{\texttt{true}}(0)$; a temporal GDL description up to time step $\widehat{n}$, $G_{\widehat{n}}$; the necessary requirements concerning legal moves, $P_{\widehat{n}-1}^{legal}$; and an encoding for each of the formulas in Φ, $\bigcup_{\varphi\in\Phi} Enc(\varphi)$. Encoding $Enc(\varphi)$ ensures that if $\eta(\varphi)$ occurs in each answer set for $P_\Phi^{bc}(G)$, then every state sequence starting at S_{init} makes φ true—which means that $S_{init} \vDash_t \varphi$.

For the *induction step* answer set program, instead of the state encoding $S_{init}^{\texttt{true}}(0)$ we need a "state generator" program whose answer sets correspond exactly to the reachable states of a GDL description. These, however, cannot be calculated efficiently in most cases, motivating an easily obtainable approximation which comprises some non-reachable states as well. The simplest approximation is the program `0{true(F,0) : fdom(F)}.`, which, together with stratified clauses defining the domain of features (`fdom`), generates *all* states. Assuming a (probably more informed) state generator S^{gen}, the *induction step* answer set program is

$$P_\Phi^{is}(G) = S^{gen} \cup G_{\widehat{n}+1} \cup P_{\widehat{n}}^{legal} \cup \bigcup_{\varphi \in \Phi} Enc(\varphi \supset \bigcirc\varphi)$$

Besides the state generator instead of the initial state, $P_\Phi^{is}(G)$ deviates from $P_\Phi^{bc}(G)$ in that the maximal time step $\widehat{n}$ is increased by one. Moreover, encoding $Enc(\varphi \supset \bigcirc\varphi)$ ensures that if $\eta(\varphi \supset \bigcirc\varphi)$ occurs in each answer set for $P_\Phi^{is}(G)$, then φ is entailed by each direct successor of a state that itself entails φ.

[3] In the following we use two common additions to ASP [8]: a *weight atom* $m\ \{\, p : d(\overline{x}) \,\}\ n$ means that for atom p an answer set has at least m and at most n different instances that satisfy $d(\overline{x})$. If n is omitted, there is no upper bound. A *constraint* is a rule `:-` $b_1, \ldots, b_k$, which excludes any answer set that satisfies $b_1, \ldots, b_k$.

4 Expressiveness and Correctness of the Proof Method

We will now show that our generalisation is correct and that it is at least as strong as the original Temporal Proof System [13]. We require two results, the first of which concerns sequences that are longer than the degree of the formula to be proved. The result refers to the standard restriction to *playable* GDL games, meaning that every role has at least one legal move in every non-terminal reachable state [7].

Lemma 1. *Let φ be a formula with degree n and G be a GDL description, then for all $\widehat{n} \geq n$:*

(A) Every $\widehat{n}$-max sequence which does not satisfy φ can be reduced to an n-max sequence which does not satisfy φ.

(B) Let G be playable and S reachable. Then every n-max sequence starting in S which does not satisfy φ can be extended to an $\widehat{n}$-max sequence starting in S which does not satisfy φ.

Note that item (B) is not true for non-reachable states S_0. Consider, e.g., formula $\varphi = \texttt{true(f)}$ in a single-player game where `f` is true initially, where the only action `a` is legal if `f` holds, and where `a` makes `f` true in the direct successor state. Assume the (non-reachable) empty state $\{\}$ to be non-terminal. Then sequence $(\{\})$ of length 0 does not satisfy φ but cannot be extended to any 1-max sequence, as the only action `a` is not legal in $\{\}$.

Our second lemma relates answer set programs to sequence-encoding stratified programs (which in turn relate to formula entailment via Definition 4).

Lemma 2. *For a GDL description G, let $P = S_0^{\texttt{true}}(0) \cup G_n \cup P_{n-1}^{legal}$. Then P has an answer set $\mathcal{A}_n$ iff there is an n-max sequence $(S_0 \stackrel{A_0}{\rightarrow} \ldots \stackrel{A_{m-1}}{\rightarrow} S_m)$ such that replacement of all rules of the form (c_3) and (c_4) in P (occurring in P_{n-1}^{legal}) with $\bigcup_{i=0}^{m-1} A_i^{\texttt{does}}(i)$ yields a program with unique standard model $\mathcal{A}_n$.*

Correctness can now be established as follows.

Theorem 1. *Let $\varphi \in \Phi$ and G be a playable GDL description with initial state S_{init}. If every answer set for $P_\Phi^{bc}(G)$ contains $\eta(\varphi)$ and every answer set for $P_\Phi^{is}(G)$ contains $\eta(\varphi \supset \bigcirc\varphi)$, then for all finite sequences $S_{init} \stackrel{A_0}{\rightarrow} S_1 \ldots \stackrel{A_{k-1}}{\rightarrow} S_k$ we have $S_k \models_t \varphi$.*

Proof. (Sketch) Induction on k, using Lemma 1 (B) and Lemma 2. Base case $k = 0$: $S_{init} \not\models_t \varphi$ implies the existence of an answer set for $P_\Phi^{bc}(G)$ that does not contain $\eta(\varphi)$. Induction step: $S_k \models_t \varphi$, $S_k \stackrel{A_k}{\rightarrow} S_{k+1}$, and $S_{k+1} \not\models_t \varphi$ imply the existence of an answer set for $P_\Phi^{is}(G)$ which does not contain $\eta(\varphi \supset \bigcirc\varphi)$.

To show that our proof method is a generalisation of the original approach, we need to restate the programs $P_\varphi^{bc}(G)$ and $P_\varphi^{is}(G)$ [13], where φ has degree n:

$$P_\varphi^{bc}(G) = S_{init}^{\texttt{true}}(0) \cup G_n \cup P_{n-1}^{legal} \cup Enc(\varphi) \cup \{\texttt{:- } \eta(\varphi)\texttt{.}\}$$

$$P_\varphi^{is}(G) = S^{gen} \cup G_{n+1} \cup P_n^{legal} \cup Enc(\varphi) \cup Enc(\bigcirc\varphi) \cup \{\texttt{:- not } \eta(\varphi)\texttt{.}, \texttt{:- } \eta(\bigcirc\varphi)\texttt{.}\}$$

The main difference is the reduced maximal time level $n \leq \widehat{n}$. Moreover the encoding for φ in $P^{bc}_{\varphi}(G)$ is constrained such as to only allow answer sets that represent φ-violating sequences. Similarly, answer sets for $P^{is}_{\varphi}(G)$ represent sequences $(S_0, \ldots, S_m)$ where φ holds in S_0 but not in S_1. Both $P^{bc}_{\varphi}(G)$ and $P^{is}_{\varphi}(G)$ being inconsistent yields $S \vDash_t \varphi$ for all reachable states.

Theorem 2. *Let $\varphi \in \Phi$ and G be a GDL description.*

- *If $P^{bc}_{\varphi}(G)$ is inconsistent then $\eta(\varphi)$ is in all answer sets of $P^{bc}_{\Phi}(G)$.*
- *If $P^{is}_{\varphi}(G)$ is inconsistent then $\eta(\varphi \supset \bigcirc\varphi)$ is in all answer sets of $P^{is}_{\Phi}(G)$.*

Proof. (Sketch) If there is an answer set for $P^{bc}_{\Phi}(G)$ ($P^{is}_{\Phi}(G)$) that does not contain $\eta(\varphi)$ ($\eta(\varphi \supset \bigcirc\varphi)$) then program transformations using Lemma 1 (A) and Lemma 2 imply that there is an answer set for $P^{bc}_{\varphi}(G)$ ($P^{is}_{\varphi}(G)$).

It should be stressed that the converse of Theorem 2, however, does not hold: An answer set for $P^{is}_{\varphi}(G)$ represents an established n-max sequence *Seq* (cf. Lemma 2) which violates $\varphi \supset \bigcirc\varphi$. *Seq* however might not be extendable to an $\widehat{n}$-max sequence (cf. the remark following Lemma 1 (B)) which could serve as counter example for $\varphi \supset \bigcirc\varphi$ in $P^{is}_{\Phi}(G)$. Hence our generalisation strengthens the result, depending on the maximal degree $\widehat{n}$ of the given formula set Φ.

5 Experimental Results

We have implemented our proof method using Fluxplayer [11] for the generation of the ASP program, which is then processed by grounder Bingo and ASP solver Clasp from a state-of-the-art answer set solving collection [10]. We use option "cautious reasoning" for Clasp to compute the intersection of all answer sets. In the following we sketch the formula sets we had the player try to prove. The resulting proof times for a variety of games can be seen in Figure 2.

- *Persistence (Φ^p)*: Ground features $f(\overline{t})$ which stay true [false] once they become true [false] are proved using the set Φ^p of *all* formulas of the form $[\neg]true(f(\overline{t})) \supset \bigcirc[\neg]true(f(\overline{t}))$. In the game Quarto, say, $\neg true(pool(X)) \supset \bigcirc\neg true(pool(X))$ can be proved for all instances $X \in \{p0000, \ldots, p1111\}$, stating that once a piece is not available for selection anymore, it will not be available throughout the remainder of the game.
- *Existence (Φ^{ex})*: We prove [non]existence of ground instances for each feature f_i/k_i and its interaction with ground instance existence of different features f_j/k_j (fixing $i < j$ in an arbitrary total feature order). The set Φ^{ex} of existential formulas contains all formulas of the form $[\neg]\varphi_{f_i/k_i}$ and all formulas of the form $\varphi_{f_i/k_i} \vee [\neg]\varphi_{f_j/k_j}$ and $\neg\varphi_{f_i/k_i} \vee [\neg]\varphi_{f_j/k_j}$, where $\varphi_{f/k} := \bigvee_{\overline{t} \in (D_1 \times \ldots \times D_k)} true(f(\overline{t}))$ (the *finite* sets $D_i \subseteq \Sigma$ being calculated automatically). For Quarto, the prover successfully shows $\neg\varphi_{slctd/1} \vee \varphi_{pctrl/1}$, hence a selected piece always implies a player to have place control. Formulas $\varphi_{pctrl/1} \vee \varphi_{sctrl/1}$ and $\neg\varphi_{pctrl/1} \vee \neg\varphi_{sctrl/1}$ prove mutual exclusion of the two control features and, together with $\neg\varphi_{pctrl/1} \vee \varphi_{cell/3}$ and $\neg\varphi_{sctrl/1} \vee \varphi_{cell/3}$, imply existence of a cell instance in each reachable state.

Game	Φ^p	Φ^{ex}	Φ^{ctrl}	$\Phi^p \cup \Phi^{ex} \cup \Phi^{ctrl}$
3pttc	0.78 (77/362)	0.45 (10/18)	0.39 (3/9)	1.55 (90/389)
bidding-tictactoe	0.18 (9/108)	0.31 (13/50)	0.23 (0/12)	0.51 (29/170)
breakthrough	1.02 (32/260)	0.78 (5/8)	1.17 (4/6)	1.69 (41/274)
capture_the_king	33.01 (7/1744)	9.65 (5/32)	29.98 (10/12)	85.05 (22/1788)
catcha_mouse	1.34 (359/998)	1.05 (8/18)	0.20 (4/6)	2.50 (371/1022)
checkers	50.47 (41/1098)	10.16 (13/32)	56.79 (4/6)	98.26 (58/1136)
chomp	0.09 (58/120)	0.14 (6/18)	0.12 (10/12)	0.20 (75/150)
connect4	0.30 (294/508)	0.32 (5/8)	0.19 (4/6)	0.73 (303/522)
endgame	453.48 (2/546)	4.54 (12/18)	33.21 (4/6)	520.80 (18/570)
knightfight	3.91 (0/608)	1.07 (2/18)	3.18 (4/12)	12.35 (6/638)
othello	3.89 (8/260)	1.41 (5/8)	4.00 (4/6)	10.34 (17/274)
pawn_whopping	0.45 (32/260)	0.20 (5/8)	0.22 (4/6)	0.74 (41/274)
quarto	38.74 (32/616)	34.48 (6/50)	33.19 (4/12)	147.02 (42/678)
tictactoe	0.09 (27/58)	0.10 (5/8)	0.13 (4/6)	0.14 (36/72)
tttcc4	15.66 (311/1244)	2.64 (7/18)	3.90 (3/9)	42.48 (321/1271)

Fig. 2. Property proof times, in seconds (average over 10 runs), for a variety of games taken from *www.general-game-playing.de*. Each time indicates one proof attempt (one ASP proof for the base case and one ASP proof for the induction step) of the respective formula set. The numbers in parantheses mean: (number of proved properties/size of the formula set). Experiments were run on an Intel Core 2 Duo CPU with 3.16 GHz.

– *Control* (Φ^{ctrl}): The periodic return of control features is proved via the set Φ^{ctrl} of all formulas $true(f(r)) \supset \bigcirc^n true(f(r))$, where r is a role and $2 \leq n \leq 4$. In Quarto we obtain successful proofs for $n = 4$ and $f(r) \in \{pctrl(white), pctrl(black), sctrl(white), sctrl(black)\}$, indicating the return of the same game phase every 4 steps.

In general, our timings for *Control* and *Persistence* are of the same order as the runtimes for games and property instances we obtained for the original method [13], since attempting proofs for all instances in one run spares the solver to repeat similar processes multiple times. This amounts to a significant speedup, which e.g. for *Persistence* means to check several hundred instances. Properties like *Existence* and *Persistence* together with initially true [false] features provide valuable information about reachable states, due to the fast timings their proofs qualify as basis for further state generator restriction, thus obtaining better timings and more accurate results for increasingly sophisticated properties. Joint proof attempts for multiple classes of properties (cf. column $\Phi^p \cup \Phi^{ex} \cup \Phi^{ctrl}$ in Figure 2), however, sometimes decrease performance (e.g. for *tttcc4*) due to less viable program rule optimisations, which suggests to divide properties in classes of "similar" form. Note that sometimes more formulas are proved (e.g. for *bidding-tictactoe*) with the joint approach thanks to the effect mentioned at the end of Section 4.

6 Summary

A key to success in general game playing is the ability to automatically infer properties of a new game that follow from the rules without being explicitly given. By extending a recently developed, basic approach to automated theorem proving for this purpose, we have developed a method that enables a general game player to systematically and simultaneously search large sets of candidate formulas in order to identify those whose validity can be established. We have formally proved the correctness of this extended method, and we have conducted systematic experiments with a variety of games that have been used by the scientific community in the past. As the experimental data show, our extended method allows to search through large sets of formulas of a similar form in times comparable to proving just a single one of these properties using the original method.

Acknowledgement. Michael Thielscher is the recipient of an Australian Research Council Future Fellowship (project number FT 0991348).

References

1. Apt, K., Blair, H.A., Walker, A.: Towards a Theory of Declarative Knowledge. In: Foundations of Deductive Databases and Logic Programming, pp. 89–148 (1987)
2. Björnsson, Y., Finnsson, H.: CADIAPLAYER: A Simulation-Based General Game Player. IEEE Transactions on Computational Intelligence and AI in Games 1(1), 4–15 (2009)
3. Clune, J.: Heuristic Evaluation Functions for General Game Playing. In: AAAI, pp. 1134–1139 (2007)
4. Gelfond, M.: Answer Sets. In: Handbook of Knowledge Representation, pp. 285–316. Elsevier, Amsterdam (2008)
5. Kuhlmann, G., Dresner, K., Stone, P.: Automatic Heuristic Construction in a Complete General Game Player. In: AAAI, pp. 1457–1462 (2006)
6. Lloyd, J., Topor, R.: A Basis for Deductive database Systems II. J. of Logic Programming 3(1), 55–67 (1986)
7. Love, N., Hinrichs, T., Haley, D., Schkufza, E., Genesereth, M.: General Game Playing: Game Description Language Specification. Technical Report, LG–2006–01, Stanford University (2006), games.stanford.edu
8. Niemelä, I., Simons, P., Soininen, T.: Stable Model Semantics of Weight Constraint Rules. In: Gelfond, M., Leone, N., Pfeifer, G. (eds.) LPNMR 1999. LNCS, vol. 1730, pp. 317–331. Springer, Heidelberg (1999)
9. Pell, B.: Strategy Generation and Evaluation for Meta-Game Playing. Ph.D., Cambridge (1993)
10. Potassco, Potsdam Answer Set Solving Collection (2008), potassco.sourceforge.net
11. Schiffel, S., Thielscher, M.: Fluxplayer: A Successful General Game Player. In: AAAI, pp. 1191–1119 (2007)
12. Schiffel, S., Thielscher, M.: Automated Theorem Proving for General Game Playing. In: IJCAI, pp. 911–916 (2009)
13. Thielscher, M., Voigt, S.: A Temporal Proof System for General Game Playing. In: AAAI, pp. 1000–1005 (2010)

Temporal Belief-Change: κ-functions Approach

Armin Hezart, Abhaya Nayak, and Mehmet A. Orgun

Department of Computing, Macquare University, Sydney, Australia
{ahezart,abhaya,mehmet}@science.mq.edu.au

Abstract. Current belief change literature is largely confined to atemporal belief change – the temporal element of beliefs is not explicitly recognized or represented. In this paper, we present a temporal belief change framework that is based on applying Spohn's theory of ranking functions to certain temporal semantic objects that we call 'histories'. The resulting framework allows us to address a class of problems for which Jeffery's *general conditionalization*, and Spohn's *cardinality of the ranks*, as well as the *dependencies between beliefs* play a central role. This allows us to lend further support to the argument that the application of the AGM theory is not necessarily limited to a static world. We also present an interpretation of *belief update* in the context of ranking-functions that has been missing in the literature.

1 Introduction

Research in the area of belief change is primarily concerned with modeling of how a rational agent should realign her beliefs in light of new information. The fountainhead of one branch of studies continues to be the AGM theory [1]. Belief update is another alternative theory where the new information is taken to indicate a change in the world [2]. Subsequently, belief update is portrayed as a model of belief change regarding a dynamic world and the AGM theory a model of belief change about a static world. Following this distinction, belief update motivated new inquiries into belief change about a dynamic world. Most of the enquires follow the underlying intuition that the world changes along the lines of transitions, where possible worlds that are considered plausible, evolve independently, to a new set of possible worlds [3].

On the other hand, a number of researchers have begun questioning the drawn 'static vs. dynamic world' distinction drawn between the AGM theory and the belief update, and argued that it is the language in AGM theory that is static and not the underlying assumption that the world is static [4,5]. It has, therefore, been proposed that "a better understanding of belief change can be gained from examining appropriate semantic models" [4]. The multiagent logic of knowledge and plausibility in time by Friedman and Halpern is the most notable of such approaches [4,6]. This paper takes a similar position, representing the temporal element of the beliefs explicitly. Our motivation is to address a class of problems that require novel notions such as the dependency relation among beliefs. These features are by and large captured in Spohn's theory of ranking functions (also called κ-functions) [7]. We motivate our work by the following example that shows that neither the AGM theory nor the belief update theory is sufficient to address a certain class of problems in belief change.

J. Li (Ed.): AI 2010, LNAI 6464, pp. 11–21, 2010.

Example 1. Doctors who diagnose patients based on the patients' description of their problems often make educated guesses about the correct symptoms. The symptoms are therefore described not as a single proposition but as a number of propositions with different strengths attached to each of them. Moreover, the identified symptoms may need to be revised, pending further tests. To emphasize this point we assume that the patient is a migrant who can communicate with the doctor only through an interpreter.

The patient 'X' pointing to her stomach tells the doctor that her stomach and abdominal area hurts badly. The doctor describes the symptoms at time t_1 by two distinct symptoms s_1, s_2 such that s_1 is surely present, s_2 is apparent, and that both the s_1 and s_2 are likely to be present. The symptoms being newly discovered are the new information. The belief change approach that allows for such a specific representation of the new information is by R. C. Jeffrey where the new information is expressed as a probability distribution over a number of propositions [8] (Chapter 11). This representation also allows for the new information to be revisable. Although, Jeffery's approach to belief change is probabilistic, his method is adopted in Spohn's theory of ranking functions which is a deterministic/qualitative theory of belief change [7].

In general, the current belief and the new belief refer to the belief prior to and after the belief change operation. Moreover, belief change operations are due to incorporating the new information that can be an observation or an action. To account for the new symptoms the doctor needs to change her current belief. The required belief change can be attained by applying either the AGM revision operation, or the belief update operation, or the conditionalization in the theory of κ-functions.

In making a diagnosis, the doctor should decide what illnesses (the new or the old illnesses) best correspond to the symptoms as there may be a number of illnesses with similar symptoms. However, neither the belief update theory nor the AGM theory can provide this required reasoning. In belief update, there are no strengths attached to the beliefs, hence, belief update can only identify the possible illnesses and not the most relevant ones. There are extensions of belief update that employ ranking; we briefly touch on two in the section 5 [9,10]. In the AGM theory, beliefs have strengths associated with them, called epistemic entrenchment values. Although, the AGM revision can provide a distinction about the strongest believed illnesses, it cannot identify the most relevant illnesses in relation to the identified symptoms. The reason is that to establish the most relevant illness, we need to compare the differences of the strength of the believed illnesses before and after the inclusion of the new symptoms. The larger a shift in the strength of a belief is, the more relevant the belief is to the new information. The entrenchment values are ordinal, and in the absence of a metric for comparing the shifts, the entrenchment values cannot be used for such purpose [9,11].

However, in Spohn's ranking theory the required reasoning is possible, because, the relevance relation among beliefs is defined, and, the strengths of beliefs are of cardinal nature. The same issue of the relevance arises when the doctor tries to determine the most effective medication. It can be seen that in regard to the class of problems where the 'relevance' and the 'comparison of the shifts' in strengths of beliefs are key issues the theory of κ-functions is the most suitable belief change theory. However, in the theory of κ-functions, the temporal element of belief is not explicitly represented.

The doctor identifies the illness c_1 and advises the medication m_1 at time t_2 and expects the patient to recover by time t_3. However, at the re-examination, time t_3, the doctor is surprised to find that the symptoms s_1 and s_2 are still present. Administering the medication is a new information, hence, the doctor again needs to change her belief. The current belief now holds that c_1, s_1 and s_2 are believed. It is established in the literature that the typical application of the AGM theory does not work in this situation [2]. The application of the belief update is problematic as well. The doctor's belief after administering the medication at t_2 should include the medication m_1 and can either include the symptoms s_1, s_2 or not, depending on how we model the update operation. We need to remember that the model of belief change should reflect our expectation that the medication comes into effect at t_3, and not instantly at t_2, and, the correct belief change after the re-examination should indicate that either the symptoms or the illnesses are identified incorrectly.

If we model the update operation such that the new belief includes s_1, s_2, then, at t_3, the doctor should not change her belief while the correct expectation requires a change in her belief. On the other hand, if the new belief does not include s_1, s_2 then it cannot correspond to t_2. If the belief corresponds to t_3 then upon learning a new information at a time t where $t_2 < t < t_3$, the doctor has no means of changing her belief at t, because her current belief at t is about the future at t_3. We therefore conclude that neither the typical application of the AGM theory nor the belief update theory is the right answer for the problem here. ■

In addition, for this example, it is evident that to express how events unfold, we require a representation of belief where the element of time is explicitly captured. In this paper, we present a temporal model for beliefs called histories that are semantic objects analogous to the models in PLTL (Propositional Linear Temporal Logic) [12]. We then develop a temporal belief change framework which is an adaptation of the theory of κ-functions to the histories. In this manner we adopt all the desired features of the ranking functions for the required temporal reasoning. Like Spohn's theory of κ-functions, the approach in this paper is purely semantic.

This paper is structured as follows. In section 2, we provide the temporal belief change framework followed by a solution to the example given above. In section 3, we discuss the relation between the presented belief change method and the AGM revision by following Spohn's interpretation of the traditional AGM revision in the ranking theory [7,11]. While the relationship between Spohn's belief change theory and the AGM theory is well investigated, a parallel analysis for the belief update is missing. In section 4, we provide a reading of the KM belief update and the iterated update in the presented framework and identify the class of temporal κ-functions that are suitable for the iterated update. Finally, in section 5, we compare our work with the works in [6,9,10,13,14], and discuss the future research directions.

2 Temporal Ranking Function: Temporal Belief Framework

In this section, we introduce the temporal belief change framework. The framework is based on ranking of semantic objects analogue to the models in PLTL called histories where ranks reflect implausibility of histories. The formal presentation of the temporal

framework is as follows. Let $\mathcal{AP}$ be a finite set of propositional variables. In usual way, possible worlds are defined as $w : \mathcal{AP} \rightarrow \{0, 1\}$, and, $\mathcal{W}$ as the complete set of possible worlds. We assume the set of natural numbers $\mathcal{N}$ as the time domain. A history h is a function that assigns to each time point $t \in \mathcal{N}$ a member of $\mathcal{W}$, $h : \mathcal{N} \rightarrow \mathcal{W}$. The complete set of histories is denoted by $\mathcal{H}$. Every history gives a complete picture of a dynamic world that is analogue to the possible worlds that give a complete picture of a static world. In fact, in relation to representing beliefs, we use possible histories the same way we use possible worlds, e.g. the example given in caption of figure 1. In addition, to represent beliefs at certain instants or intervals we define:

1. For $A \subseteq \mathcal{W}$ and $t \in \mathcal{N}$: $(A, t) \triangleq \{h \in \mathcal{H} \mid h(t) \in A\}$. That is (A, t) is the set of histories that pass through any world $w \in A$ at t. For singleton sets, we write (w, t) instead of $(\{w\}, t)$.
2. By a Sub-history $\langle (w_0, t), \cdots (w_m, t + m) \rangle$, we mean $(w_0, t) \cap \cdots \cap (w_m, t + m)$ where $w_0 \cdots w_m \in \mathcal{W}$. A sub-history can be open from the right hand side.

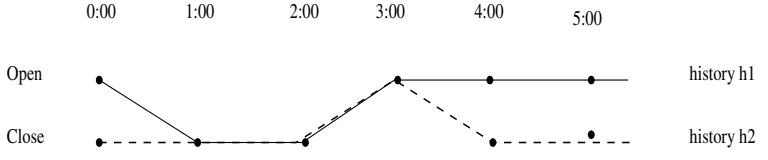


Fig. 1. The two histories h_1, h_2 show the possible states of a door being Open or Closed at different hours: a represents "The door is open", $\mathcal{AP}$ = $\{a\}$. 'Open' and 'Closed' represent the two possible worlds w_1, w_2: $w_1(a)$ = 1, $w_2(a)$ = 0. Assuming that only histories h_1, h_2 are plausible then the agent believes: (Close, 1) as well as, the sequence $\langle$(Close , 1), (Close , 2), (Open , 3)$\rangle$.

Next, we apply Spohn's κ-functions to histories. The theory of κ-functions is a framework for expressing belief kinematics [7]. κ-functions are presented as the qualitative or deterministic counterpart of probability space representing degrees of disbelief. In κ-functions, possible worlds (alt. propositions) are ranked according to their implausibility or degree of surprise. The degree of surprise associated with a proposition A is roughly equal to how least surprising A can possibly be. If $\kappa(A)$ = 0 then A is considered plausible, otherwise, A is implausible. A proposition A is believed only if $\bar{A}$ is implausible. κ-functions are adapted to histories as follows. We treat κ-functions as abstract measures that rank semantic objects, and then substitute set of histories H for propositions A. In this manner, instead of ranking propositions we rank set of histories (or alternatively, instead of ranking possible worlds we rank histories):

Definition 1. *(Spohn 2005) Let* $\mathbf{H} \subseteq 2^{\mathcal{H}}$ *be a complete field, and* $H, H_1, H_2 \in \mathbf{H}$*, then a ranking function* κ *for* $\mathbf{H}$ *is a function from* $\mathbf{H}$ *to* $\mathcal{R}^*$*, where* $\mathcal{R}^* = \mathcal{R}^+ \cup \{\infty\}$*, that satisfies the following:*

1. $\kappa(H) \geq 0$*,* $\kappa(\mathcal{H}) = 0$ *and* $\kappa(\emptyset) = \infty$
2. $\kappa(H_1 \cup H_2) = min(\kappa(H_1), \kappa(H_2))$

3. The conditional rank of H_2 given H_1 is defined as: $\kappa(H_2|H_1) = \kappa(H_2 \cap H_1) - \kappa(H_1)$ if $H_1 \cap H_2 \neq \emptyset$, otherwise $\kappa(H_2|H_1) = \infty$.
4.The core-state E is: $E = \{h \in \mathcal{H} \mid \kappa(\{h\}) = 0\}$.

Defining expressions in terms of histories allows us to derive their κ in a seamless manner. For instance, we say that we believe A at time t if and only if all histories that pass through $\bar{A}$ at t are implausible, i.e. $\kappa((\bar{A}, t)) > 0$. We may now briefly discuss the choice of ranking histories against few alternative options.

One alternative method is to rank the objects (A, t) or (w, t). However, in this representation, the link between past, present and future is not explicitly represented. Another alternative approach is to rank transitions. A transition can be defined as a tuple (w, w'). The typical approach is to apply Markov assumption [3]. The general approach is to consider a preference ordering over sequences of transitions where the ordering is derived from the rankings of transitions. However, a sequence of transitions is a different representation of a history, a transition based history. Hence, we will again be ranking histories. Yet, the choice of ranking histories is best justified by (1) the representational efficacy of linear temporal models in traditional temporal logics [12], and (2) the probability measures over runs in probabilistic reasoning that is analogue to the ranking of histories.

The standard belief change method in theory of κ-functions is called conditionalization. Conditionalization is modification of κ to account for the new information.

Definition 2. *(Spohn2005) Let κ be a ranking function for $2^{\mathcal{H}}$, $H \in 2^{\mathcal{H}}$ and λ a ranking function over $\mathbf{H}$ a complete subfield of $2^{\mathcal{H}}$, then the conditionalization of κ by λ is defined as: $\kappa_\lambda(H) = \min(\{\lambda(X)+\kappa(H|X) \mid X$: atom of $\mathbf{H}$, $\kappa(X) \neq \infty\})$.*

The two features of κ-functions that Spohn discusses frequently are the relevance relation among beliefs and the cardinal nature of the ranks [7]. The relevance relation among beliefs represent the degrees which beliefs support or weaken each other [11,7]. The cardinality of ranks allows us to compare the degrees of relevance among beliefs, as well as the strength of beliefs before and after conditionalization.

Definition 3. *(Spohn 2005) Let κ be a ranking function for $2^{\mathcal{H}}$, and $H_1, H_2, H_3 \subseteq \mathcal{H}$. Then H_1 is said to be a reason for H_2, or positively relevant to H_2, given H_3 w.r.t. κ iff $\kappa(H_2 \mid H_1 \cap H_3) < \kappa(H_2 \mid \bar{H}_1 \cap H_3)$. H_1 is to be a reason against H_2, or negatively relevant to H_2, given H_3 w.r.t. κ iff $\kappa(H_2 \mid H_1 \cap H_3) > \kappa(H_2 \mid \bar{H}_1 \cap H_3)$. Finally, H_1 is irrelevant to or independent of H_2 given H_3 w.r.t. κ iff H_1 is a neither a reason for nor against H_2 given H_3 w.r.t. κ.*

The definition above allows us to make statements such as the event (A, t_A) is a reason for the event (B, t_B), given a circumstance (C, t_C) where $t_A \leq t_C < t_B$, or a number of events $((A_1, t_{A_1}) \cap (A_2, t_{A_2}) \cap (A_3, t_{A_3}))$ are a reason for the event (B, t_B) [7]. As explained in the introduction, the features 'relevance among beliefs' and 'the cardinality of the ranks' play an important role in addressing the problem in example 1. Next, we provide a solution for the example 1 using the presented framework.

Example 1 continued. Let t_i be a variable for time, and S_i, C_i, M_i be accordingly the propositions for 'symptom s_i is present', 'illness c_i is present', and 'medication m_i is present'. We also assume that the doctor is the meta reasoner that makes judgments and

decisions, and gives the input to and reads the output from the automated reasoner. The the automated reasoner is presented temporal framework.

(1) Doctor states the described symptoms at time t_1 by a ranking function λ over the subfield $\langle \emptyset, (S_1 \cap S_2, t_1), (S_1 \cap \overline{S_2}, t_1), (\overline{S_1} \cap S_2, t_1), (\overline{S_1} \cap \overline{S_2}, t_1), \mathcal{H} \rangle$ where $(S_1 \cap S_2, t_1)$ is believed. The ranking function λ is the new information to the reasoner.
(2) Reasoner conditionalizes its ranking function based on λ and gives the output that (C_1, t_1) is the most positively relevant illness to both the (S_1, t_1) and (S_2, t_1).
(3) Doctor decides that C_1 is the illness and asks the reasoner for the best medication at $t_2 > t_1$. The best medication (M_1, t_2) is the medicine M_1 such that conditionally on (M_1, t_2), (S_1, t_3) and (S_2, t_3) are the most negatively relevant to (C_1, t_1) where $t_3 > t_2$. That is the medication (M_1, t_2) will relieve the symptoms (S_1, t_3) and (S_2, t_3).
(4) Doctor reexamines the patient at t_3 and finds that S_1 and S_2 are still present.
(5) Reasoner conditionalizes by $(S_1 \cap S_2, t_3)$ and outputs that (C_1, t_1) is now disbelieved. Keeping in mind that $(S_1 \cap S_2, t_3)$ was negatively related to (C_1, t_1).
(6) Doctor orders a new blood test that shows the correct symptom is (S_3, t_4).
(7) Reasoner conditionalizes based on (S_3, t_4). Consequently: $(S_1 \cap S_2, t_1)$, the information at t_1 in step (1), is no longer believed, (C_1, t_1) is now disbelieved, and $(C_2, t_1) \cap (C_2, t_2) \cap (C_2, t_3) \cap (C_2, t_4)$ is believed where C_2 is a new illness.
(8) Repeating the steps (2) and (3), the medication (M_2, t_5) is advised and administered where the symptoms and the illness recede by time $t_6 > t_5$. ■

3 Temporal κ and AGM Revision

In regard to drawing a reading of the AGM belief change in our framework, we do not state or discuss the AGM postulates by themselves, instead, we follow Spohn's analysis. Spohn in [7,11] shows that the usual special case conditionalization $\kappa_{A,r}$ where κ is for $2^{\mathcal{W}}$, $A \in 2^{\mathcal{W}}, r \in \mathcal{R}^+$, satisfies all the AGM postulates semantically. Spohn first defines an AGM style revision function $*(A)$, and then shows that the induced revision function from $\kappa_{A,r}$ belongs to this class of AGM style revision functions.

As mentioned in the introduction, the AGM belief change is typically associated with the belief change regarding a static world. The key to this distinction is whether or not the new information indicates a change in the world. However, how an agent should determine whether or not the new information indicates a change in the world, has been neither discussed nor mentioned in the literature. In order to circumvent this difficult issue, by a 'static world' we mean " the belief state is synchronized with the received information when the belief change is performed". Under this reading the typical AGM revision yields the same result as the normal conditionalization approach presented here, shown by the following observation that says the conditionalization commutes over time t .

Definition 4. *Given a ranking function κ for $2^{\mathcal{H}}$, an induced ranking function κ^t for $2^{\mathcal{W}}$ at $t \in \mathcal{N}$ is defined as $\kappa^t(A) \triangleq \kappa((A, t))$, where $A \in 2^{\mathcal{W}}$. In addition, we define E^t, the core state at t as $E^t \triangleq \{w \in \mathcal{W} \mid \kappa^t(w) = 0\}$. Furthermore, we extend this notation to the κ and the core state after the conditionalization: $\kappa^t_{H,r}$ and $E^t_{H,r}$.*

Observation 1. *If κ is a ranking function for $2^{\mathcal{H}}$, and (A,t) the new information, $A \in 2^{\mathcal{W}}$, and $r \in \mathcal{R}^+$ then $(\kappa^t)_{A,r} = \kappa^t_{(A,t),r}$.*

In AGM theory, the language representing belief is typically propositional while the AGM does not make such distinction. Hence, given the appropriate semantic representation of belief where the temporal element is explicitly stated, the AGM theory can also be a belief change theory regarding a dynamic world [4]. We follow this argument and propose that an AGM theory where the language is PLTL (with the consequence relation $\vdash_{PLTL}$,) is a belief change about a dynamic world. We substantiate this claim as follows. In section 2, we remarked that histories are analogous to the models in PLTL. Let us apply the κ-functions to the models in PLTL similar to how we applied κ-functions to the histories. Then, following the Spohn's method by which he shows his usual conditionalization function is analogue to the AGM revision operation, we can show that the temporal conditionalization presented here is an AGM compatible belief change operation [7,11]. We do not show this claim formally due to (1) the space constraint and (2) that the traditional temporal logics are designed to express the general properties of a system where the specific instants are not relevant. However, in our theory of belief change, each piece of information corresponds to specific time points. Next, we look at the KM belief update under the presented temporal settings.

4 Temporal $\mathcal{K}$ and KM Belief Update

Spohn in a number of works analyzes the relation between his theory of ranking functions and the AGM theory. However, little attention is paid to the belief update. In this section, we provide a reading for the theory of belief update. In [2] Katsuno and Mendelzon present 'belief update' as a theory of belief change when the new information is indicative of a change in the world. The syntactic update operator is characterized by a set of postulates. The semantic equivalent of the update operator is given as: $g : 2^{\mathcal{W}} \times 2^{\mathcal{W}} \rightarrow 2^{\mathcal{W}}$. The operator g selects those models of the new information that are 'most likely' to follow the models of the current belief, individually [2,9]:

$$g(\, E, A \,) = \bigcup_{w \in E} \{ w' \in A \mid \forall w'' \in A, w' \leq_w w'' \} \, .$$

The 'most likely' models are selected by a partial order $\leq_w$ that is 'faithful' to w. A partial order $\leq_w$ over $\mathcal{W}$ is faithful to w if and only if: If $w' \neq w$ then $w <_w w'$ holds ($<_w$ is derived in usual manner from $\leq_w$).

Belief update can also be viewed as a theory that draws the new belief from the current plausible worlds w, by means of the best possible transitions (w, w') [3]. Hence, the relationship $\leq_w$ describes the preferences over transitions (w, w'). In our setting, the transitions (w, w') can be represented by sub-histories $\langle (w,t), \cdots, (w', t+m) \rangle$ for some $m > 0$. The induced order $\leq_w$ will then be the κ of sub-histories $\langle (w,t), \cdots, (w', t+m) \rangle$. The induced order can however be time variant. In order to account for the faithfulness, we manually assign the transitions (w, w) the ranking zero. In order that the possible worlds evolve independently, we conditionalize the ranks with respect to the least surprising sub-histories that start from each plausible world and end with a possible model of the new information. An update conditionalization is natural when the faithfulness, i.e. the inertia, is not enforced manually.

Definition 5. *Let κ be a ranking function over $2^{\mathcal{H}}$, (A, t_A) the new information, $A \in 2^{\mathcal{W}}, t_A \in \mathcal{N}$, and $t^* < t_A$ and H^1_w, H^2_w, H_w are define as: $H^1_w = (w, t^*) \cap (w, t^* + 1) \cap \cdots \cap (w, t_A)$, $H^2_w = (w, t^*) \cap (\mathcal{W}, t^* + 1) \cap \cdots \cap (\mathcal{W}, t_A - 1) \cap (A, t_A)$, and $H_w = H^1_w$ if $w \in A$, otherwise, $H_w = H^2_w$. Then, the conditionalization $\kappa_{(A,t_A)}(H) \triangleq \min(\{\kappa(H \mid H_w) \mid w \in E^{t^*}\})$ w.r.t. time t^* is called a natural update conditionalization if $H_w = H^2_w$, and, an inertia enforced update conditionalization if $H_w = H^1_w$.*

An implicit characteristic of belief update is that the belief update accepts the current belief state [3]. To represent this feature, we define the 'past preserving' property, and show that both the update conditionalization functions above satisfy this property.

Definition 6. *Let κ be a ranking function for $2^{\mathcal{H}}$, and (A, t_A), $A \in 2^{\mathcal{W}}, t_A \in \mathcal{N}$, the new information. Then the conditionalization $\kappa_{(A,t_A)}(H)$ w.r.t. instant $t^* < t_A$ is:*
(1) past preserving iff $E^{t^*} = E^{t^*}_{(A,t_A)}$.
(2) inertia preserving iff $(E^{t^*} \cap A) \subseteq E^{t_A}_{(A,t_A)}$.

Theorem 1. *Both the natural and the inertia enforced update Conditionalizations are past preserving but only inertia enforced update is inertia preserving.*

The above theorem states that in order to attain an update operation, we need to enforce inertia, i.e. faithfulness property, manually. However, it is more appropriate for the belief attitude κ to have a structure that supports inertia automatically. Under such class of κ-functions, the natural update conditionalization can be applied repeatedly. In addition, in relation to successive updates, we extend the property 'past preserving' to 'path preserving' where a path is the plausible sub-history since the last update.

Definition 7. *Let κ be a ranking function for $2^{\mathcal{H}}$, and (A, t_A), $A \in 2^{\mathcal{W}}, t_A \in \mathcal{N}$, the new information, then the conditionalization $\kappa_{(A,t_A)}$ is said to be path preserving w.r.t. the time t^* where $t^* \leq t_0$ and t_0 is the time stamp of the last conditionalization, iff: if $\kappa(H) = 0$ (respectively > 0), then $\kappa_{(A,t_A)}(H) = 0$ (respectively > 0), for all $H = \langle (w', t'), \cdots, (w'', t'') \rangle$ where $w', w'' \in \mathcal{W}$, and, $t', t'' \in \mathcal{N}$ and $t^* \leq t' \leq t'' \leq t_A$.*

To identify the class of appropriate κ-functions, we present two structural constraints on the κ-function. The first constraint enforces the faithfulness condition over the κ functions that every static sub-history, i.e. a non-changing sub-history, is less surprising than any sub-history that intersect it. As a result, the rankings and the belief status of all propositions remain static (theorem 2 below). The second constraint is a time-variant Markovian property (a nonhomogeneous Markov property) that ensures the path preserving property holds.

Definition 8. *A ranking function κ for $2^{\mathcal{H}}$ is orderly distributed at (w, t), $w \in \mathcal{W}, t \in \mathcal{N}$, w.r.t. inertia after $t^* \in \mathcal{N}$ iff for all $H_{w_i w w_j} \triangleq \langle (w_i, t-n), \cdots, (w, t), (w_j, t+1) \rangle$ where $w_i, w_j \in \mathcal{W}$, $0 < n \leq t - t^*$, we have:*
(1) $\kappa(H_{www}) < \kappa(H_{w_i w w})$, and
(2) $\kappa(H_{www_j}) - \kappa(H_{www}) = \kappa(H_{w_i ww_j}) - \kappa(H_{w_i ww}) > 0$.

Theorem 2. *Let a given κ over $2^{\mathcal{H}}$ be orderly distributed w.r.t. inertia after $t^* \in \mathcal{N}$ for all (w, t), $w \in \mathcal{W}, t \in \mathcal{N}$. Then for all $t \geq t^*$, $\forall j \geq 0$, $\kappa(w, t) = \kappa(w, t + j) = \kappa(H_w)$ where H_w is sub-history $\langle (w, t), (w, t + 1), \cdots, (w, t + j) \rangle$.*

The two constraints above are sufficient for a κ-function not only to be both inertia and path preserving, but also to preserve its structure, after each natural update conditionalization. Conversely, a κ-function that after each natural update conditoinalization satisfies the properties above, will be orderly distributed regarding inertia. Hence, the two structural constraints above provide a 'necessary and sufficient condition' that identifies the class of κ-function for which the natural update can be applied iteratedly.

Theorem 3. *Let κ over $2^{\mathcal{H}}$ be orderly distributed w.r.t. inertia after t^* for all (w,t), $w \in \mathcal{W}$, $t > t^*$. Let also t_0 be the time stamp of last update, and (A,t_1), $A \in 2^{\mathcal{W}}$, be the new information where $t_1 > t_0 > t^*$. Then, the iterated condititionalization by natural update is both path preserving and inertia preserving, and after each conditionaliztion, the κ is still uniformly distributed w.r.t. inertia after t_1, for all (w,t), $t > t_1$.*

Theorem 4. *Let κ be a ranking function for $2^{\mathcal{H}}$ for which any iterated condititionalization by natural update after $t^* \in \mathcal{N}$ is both path and inertia preserving. Then, κ is orderly distributed w.r.t. inertia after t^* for all (w,t), $w \in \mathcal{W}, t \in \mathcal{N}$, $t > t^*$.*

5 Discussion and Conclusion

There are many works on belief change regarding a dynamic world where some apply extensions of belief update method [5]. Next, we look at five works [6,9,10,13,14]. All the five frameworks are designed to handle inductive, abductive and deductive reasoning in relation to a sequence of information. The first four works use semantic objects similar to the histories that are the sequence of static models [6,10,13,14]. The belief change methods in all the four frameworks include a filtering action that screens out those semantic objects that do not adhere to the new information (alternatively sequence of information). The First framework is called extrapolation [13]. Extrapolation draws the most 'plausible' sequence of sentences that agree with a sequence of information, called scenario. The models for scenarios are called trajectories. The second work presents an iterated belief change theory based on a sequence of consistent infallible actions and observations in an action based transition system [14].

The third work is the comprehensive framework by Friedman and Halpern where they present a general framework for agents to reason about knowledge and belief in the context of time [6]. The semantics of belief and belief change is given based on the 'plausibility measures' and the 'conditional plausibility measure' on runs [6] where runs are sequences of states. The plausibility measure is an abstract measure for uncertainty and is shown to generalize other measures for uncertainty including Spohn's κ-functions [6]. However, the plausibility measures are not of cardinal nature, due to their generality. The other significant difference between the Friedman and Halpern's framework and ours is that the conditionalization of plausibility measures does not follow Jeffrey's general conditionalization.

The filtering action in these three frameworks effectively makes the new information not-revisable. Spohn's conditionalization is more general than this type of filtering such that the effects of filtering can be emulated by making the new information infallible i.e. assigning the ranking of ∞ to the histories that do not satisfy the information. Conditionalization by an infallible H preserves the order of κ of histories after revision.

The fourth work is the BReLS system where the method of belief change selects the best sequence of static models in relation to multiple pieces of information with different degrees of reliability [10]. The selection function employs a cost function that is the sum of the cost of changes from one state to the next, and, a variation of Hamming distance between models and the most reliable information. Our framework differs with BReLs system on a number of accounts. Unlike the theory of κ-functions, in BReLS system, the distances between models and sentences are invariant with respect to time and the belief change operation. Moreover, the cost of changes depend only on the present state, indicating that the BReLS system implicitly applies the Markovian assumption. In addition, the belief change method in BReLS system is commutative with respect to the new information while in the theory of κ-functions it is not necessarily commutative.

In conclusion, neither of the four frameworks above have the specific features of the theory of ranking functions that we require. Moreover, we like to highlight that Spohn, in presenting his framework, has argued that the various features of his framework are required to account for a theory of belief change; drawing a distinction between his framework and a theory that merely employs rankings [7]. In this work, we aimed to adopt his framework in its entirety.

The last framework of interest is the generalized update that integrates the belief revision and update methods, in conjunction with possible events, to draw the best explanation for the new information [9]. The generalized update employs three Spohnian style, cardinal κ-functions for belief states, the possible events in relation to states, and, the possible outcomes of events. However, to apply the iterated generalized update, we require to employ the Markovian assumption that is akin to our future work [3]. In order to address the computational problems of the ranking of the histories, our next task is to apply the qualitative Markov assumption to the rankings of histories.

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. Journal of Symbolic Logic, 510–530 (1985)
2. Katsuno, H., Mendelzon, A.: On the difference between updating a knowledge base and revising it. In: Principles of Knowledge Representation and Reasoning, pp. 387–394 (1991)
3. Friedman, N., Halpern, J.Y.: A qualitative markov assumption and its implications for belief change. In: UAI, pp. 263–273 (1996)
4. Friedman, N., Halpern, J.Y.: Modeling belief in dynamic systems, part ii: Revisions and update. CoRR cs.AI/0307071 (2003)
5. Lang, J.: Belief update revisited. In: IJCAI, pp. 2517–2522 (2007)
6. Friedman, N., Halpern, J.Y.: Modeling belief in dynamic systems, part i: Foundations. CoRR cs.AI/0307070 (2003)
7. Spohn, W.: A survey of ranking theory. In: Huber, F., Schmidt-Petri, C. (eds.) Degrees of Belief. An Anthology. Oxford University Press, Oxford (2005)
8. Jeffrey, R.: The Logic of Decision, 2nd edn. University of Chicago Press, Chicago (1965/1983)
9. Boutilier, C.: Generalized update: Belief change in dynamic settings. In: IJCAI 1995, pp. 1550–1556 (1995)

10. Liberatore, P., Schaerf, M.: Brels: A system for the integration of knowledge bases. In: KR, pp. 145–152 (2000)
11. Spohn, W.: Ranking functions, agm style. In: Hansson, B., Halld, S., Sahlin, N.-E., Rabinowicz, W. (eds.) Internet Festschrift for Peter Gärdenfors, Lund (1999)
12. Emerson, E.: Temporal and modal logic. In: Handbook of Theoretical Computer Science, ch. 16. MIT Press, Cambridge (1990)
13. de Saint-Cyr, F.D., Lang, J.: Belief extrapolation (or how to reason about observations and unpredicted change). In: KR, pp. 497–508 (2002)
14. Hunter, A.A., Delgrande, J.P.: Iterated belief change: A transition system approach. In: International Joint Conference on Artificial Intelligence, IJCAI (2005)

Resource-Sensitive Reasoning with Sequential Information

Norihiro Kamide[1] and Ken Kaneiwa[2]

[1] Waseda Institute for Advanced Study, Waseda University, Japan
drnkamide08@kpd.biglobe.ne.jp
[2] Department of Electrical Engineering and Computer Science, Iwate University, Japan
kaneiwa@cis.iwate-u.ac.jp

Abstract. A logic called sequence-indexed linear logic (SLL) is proposed to appropriately formalize resource-sensitive reasoning with sequential information. The completeness and cut-elimination theorems for SLL are shown, and SLL and a fragment of SLL are shown to be undecidable and decidable, respectively. As an application of SLL, some specifications of secure password authentication systems are discussed.

1 Introduction

In this paper, a new logic, sequence-indexed linear logic (SLL), is obtained from Girard's *linear logic* [1] by adding a sequence modal operator, which represents a sequence of symbols. By the sequence modal operator in SLL, we can appropriately express sequential information in resource-sensitive reasoning.

The notion of "resources," encompassing concepts such as processor time, memory, cost of components and energy requirements, is fundamental to computational systems [6]. In the area of knowledge representation and reasoning, this notion is very important for handling real scheduling problems to construct complex plans of actions since many actions consume resources such as money, gas, and raw materials [7] (see Section 12 in [7]).

It is known that Girard's linear logic can elegantly represent the notion of "resources" [1]. In linear logic, the concept of "resource consumption" can be represented by using the linear implication connective $\rightarrow$ and the fusion connective $*$, and the concept of "reusable resource" can be represented by using the linear exponential operator !. A typical example formula is as follows: $coin * coin \rightarrow coffee * (!water)$. This example means "if we spend two coins, then we can have a cup of coffee and as much of water as we like" when the price of coffee is two coins and water is free. It is to be noted that this example cannot be expressed using classical logic, since the formula $coin \wedge coin$ in classical logic is logically equivalent to $coin$, i.e., classical logic is not resource-sensitive.

In order to discuss certain real and practical examples, the resource descriptions should be more fine-grained and expressive and capable of conveying additional information. For example, the following expression may be necessary for a practical situation: $[teashop\ ;\ john](coin * coin * coin \rightarrow [1min\ ;\ 1min]coffee *$

J. Li (Ed.): AI 2010, LNAI 6464, pp. 22–31, 2010.

$[1min]water$). This example means "in a teashop, if John spends three coins, then he can have a cup of coffee after two minutes and a cup of water after one minute." In this example, the expressions $[teashop\ ;\ john]$ and $[1min\ ;\ 1min]$, which are regarded as additional information, can naturally be represented by the sequence modal operator in SLL.

The reason underlying the use of the notion of "sequences" in the new modal operator is explained below. The notion of "sequences" is fundamental to practical reasoning in computer science because it can appropriately represent "data sequences," "program-execution sequences," "action sequences," "time sequences" etc. The notion of sequences is thus useful to represent the notions of "information," "attributes," "trees," "orders," "preferences," "strings," "vectors," and "ontologies". Additional (sequential) information can be represented by sequences; this is especially suitable because a sequence structure gives a *monoid* $\langle M, ;, \emptyset \rangle$ with *informational interpretation* [8]:

1. M is a set of pieces of (ordered) information (i.e., a set of sequences),
2. ; is a binary operator (on M) that combines two pieces of information (i.e., a concatenation operator on sequences),
3. $\emptyset$ is the empty piece of information (i.e., the empty sequence).

The sequence modal operator $[b]$ represents sequential information as labels. A formula of the form $[b_1\ ;\ b_2\ ;\cdots;\ b_n]\alpha$ intuitively means that "α is true based on a sequence $b_1\ ;\ b_2\ ;\cdots;\ b_n$ of (ordered or prioritized) information pieces." Further, a formula of the form $[\emptyset]\alpha$, which coincides with α, intuitively means that "α is true without any information (i.e., it is an eternal truth in the sense of classical logic)."

In this paper, a *sequence-indexed phase semantics* for SLL is introduced by generalizing Girard's phase semantics [1], and the cut-elimination and completeness theorems for SLL are shown by using a modification of Okada's phase semantic proof [5]. SLL and the !-free fragment of SLL are shown to be undecidable and decidable, respectively, by using an embedding-based method. Specifications of secure password authentication systems, which are proposed as an application of SLL, are discussed. In this application, the following expression is used: $[server\ ;\ client](error * error * (!error)) \rightarrow [server\ ;\ client]reject$ which means "if a server returns the error messages more than twice to a client, then the server returns the password reject message to the client." Note that the error messages are expressed as a "resource" by using the connectives $*$ and !, and the "information" on servers, clients, and login-attempts is expressed by the sequence modal operator.

Many studies have reported on the applications of linear logic to programming languages (e.g., [4] and the references therein). By the embedding theorem (of SLL into ILL) proposed in this paper, we can translate the set of formulas of SLL into that of ILL; hence, the proposed framework, which is based on SLL, can also be adopted to the previously established framework of linear logic programming. Therefore, some SLL-based applications are executable in the established linear logic programming frameworks. It should also be noted that the sequence modal

operator can be adapted to a wide range of non-classical logics. In this respect, Kamide and Kaneiwa [2] studied an extended *full computation-tree logic* with the sequence modal operator. They showed that the sequence modal operator is applicable to certain ontological descriptions.

2 Sequence-Indexed Linear Logic

Prior to the precise discussion, the language used in this paper is introduced below. *Formulas* are constructed from propositional variables, **1** (multiplicative truth constant), $\top$ (additive truth constant), $\bot$ (additive falsity constant), $\rightarrow$ (implication), $\wedge$ (conjunction), $*$ (fusion), $\vee$ (disjunction), ! (exponential), and $[b]$ (sequence modal operator) where b is a sequence. *Sequences* are constructed from atomic sequences, $\emptyset$ (empty sequence) and ; (composition). Lower-case letters $b, c, ...$ are used for sequences, lower-case letters $p, q,...$ are used for propositional variables, Greek lower-case letters $\alpha, \beta, ...$ are used for formulas, and Greek capital letters $\Gamma, \Delta, ...$ are used for finite (possibly empty) multisets of formulas. For any $\sharp \in \{!, [b]\}$, an expression $\sharp\Gamma$ is used to denote the multiset $\{\sharp\gamma \mid \gamma \in \Gamma\}$. The symbol $\equiv$ is used to denote the equality of sequences (or multisets) of symbols. An expression $[\emptyset]\alpha$ means α, and expressions $[\emptyset\ ;\ b]\alpha$ and $[b\ ;\ \emptyset]\alpha$ mean $[b]\alpha$. An expression Γ^* means $\Gamma^* \equiv \gamma_1 * \cdots * \gamma_n$ if $\Gamma \equiv \{\gamma_1, \cdots, \gamma_n\}$ $(0 < n)$ and $\Gamma^* \equiv \emptyset$ if $\Gamma \equiv \emptyset$. A *sequent* is an expression of the form $\Gamma \Rightarrow \gamma$ where γ is nonempty. It is assumed that the terminological conventions regarding sequents (e.g., antecedent and succedent) are the usual ones. If a sequent S is provable in a sequent calculus L, then such a fact is denoted as $L \vdash S$ or $\vdash S$. The parentheses for $*$ is omitted since $*$ is associative, i.e., $\vdash \alpha * (\beta * \gamma) \Rightarrow (\alpha * \beta) * \gamma$ and $\vdash (\alpha * \beta) * \gamma \Rightarrow \alpha * (\beta * \gamma)$ for any formulas α, β and γ.

Definition 1. *Formulas and sequences are defined by the following grammar, assuming p and e represent propositional variables and atomic sequences, resp.:*

$$\alpha ::= p \mid \mathbf{1} \mid \top \mid \bot \mid \alpha{\rightarrow}\alpha \mid \alpha \wedge \alpha \mid \alpha * \alpha \mid \alpha \vee \alpha \mid !\alpha \mid [b]\alpha$$
$$b ::= e \mid \emptyset \mid b\ ;\ b$$

The set of sequences (including the empty sequence) is denoted as SE. An expression $[\hat{d}]$ is used to represent $[d_0][d_1]\cdots[d_i]$ with $i \in \omega$ and $d_0 \equiv \emptyset$, i.e., $[\hat{d}]$ can be the empty sequence. Also, an expression $\hat{d}$ is used to represent $d_0\ ;\ d_1\ ;\ \cdots\ ;\ d_i$ with $i \in \omega$ and $d_0 \equiv \emptyset$.

Definition 2. *The initial sequents of* SLL *are of the form: for any propositional variable p,*

$$[\hat{d}]p \Rightarrow [\hat{d}]p \qquad \Rightarrow [\hat{d}]\mathbf{1} \qquad \Gamma \Rightarrow [\hat{d}]\top \qquad [\hat{d}]\bot, \Gamma \Rightarrow \gamma.$$

The inference rules of SLL *are of the form:*

$$\frac{\Gamma \Rightarrow \alpha \quad \alpha, \Delta \Rightarrow \gamma}{\Gamma, \Delta \Rightarrow \gamma}\ (\text{cut}) \qquad \frac{\Gamma \Rightarrow \gamma}{[\hat{d}]\mathbf{1}, \Gamma \Rightarrow \gamma}\ (\mathbf{1}\text{we})$$

$$\frac{\Gamma \Rightarrow [\hat{d}]\alpha \quad [\hat{d}]\beta, \Delta \Rightarrow \gamma}{[\hat{d}](\alpha{\to}\beta), \Gamma, \Delta \Rightarrow \gamma}\ (\to\text{left}) \qquad \frac{[\hat{d}]\alpha, \Gamma \Rightarrow [\hat{d}]\beta}{\Gamma \Rightarrow [\hat{d}](\alpha{\to}\beta)}\ (\to\text{right})$$

$$\frac{[\hat{d}]\alpha, [\hat{d}]\beta, \Gamma \Rightarrow \gamma}{[\hat{d}](\alpha * \beta), \Gamma \Rightarrow \gamma}\ (*\text{left}) \qquad \frac{\Gamma \Rightarrow [\hat{d}]\alpha \quad \Delta \Rightarrow [\hat{d}]\beta}{\Gamma, \Delta \Rightarrow [\hat{d}](\alpha * \beta)}\ (*\text{right})$$

$$\frac{[\hat{d}]\alpha, \Gamma \Rightarrow \gamma}{[\hat{d}](\alpha \wedge \beta), \Gamma \Rightarrow \gamma}\ (\wedge\text{left1}) \qquad \frac{[\hat{d}]\beta, \Gamma \Rightarrow \gamma}{[\hat{d}](\alpha \wedge \beta), \Gamma \Rightarrow \gamma}\ (\wedge\text{left2})$$

$$\frac{\Gamma \Rightarrow [\hat{d}]\alpha \quad \Gamma \Rightarrow [\hat{d}]\beta}{\Gamma \Rightarrow [\hat{d}](\alpha \wedge \beta)}\ (\wedge\text{right}) \qquad \frac{[\hat{d}]\alpha, \Gamma \Rightarrow \gamma \quad [\hat{d}]\beta, \Gamma \Rightarrow \gamma}{[\hat{d}](\alpha \vee \beta), \Gamma \Rightarrow \gamma}\ (\vee\text{left})$$

$$\frac{\Gamma \Rightarrow [\hat{d}]\alpha}{\Gamma \Rightarrow [\hat{d}](\alpha \vee \beta)}\ (\vee\text{right1}) \qquad \frac{\Gamma \Rightarrow [\hat{d}]\beta}{\Gamma \Rightarrow [\hat{d}](\alpha \vee \beta)}\ (\vee\text{right2})$$

$$\frac{[\hat{d}]\alpha, \Gamma \Rightarrow \gamma}{[\hat{d}]!\alpha, \Gamma \Rightarrow \gamma}\ (!\text{left}) \qquad \frac{[\hat{d}_1]!\gamma_1, ..., [\hat{d}_k]!\gamma_k \Rightarrow [\hat{e}]\alpha}{[\hat{d}_1]!\gamma_1, ..., [\hat{d}_k]!\gamma_k \Rightarrow [\hat{e}]!\alpha}\ (!\text{right}) \qquad \frac{\Gamma \Rightarrow \gamma}{[\hat{d}]!\alpha, \Gamma \Rightarrow \gamma}\ (!\text{we})$$

$$\frac{[\hat{d}]!\alpha, [\hat{d}]!\alpha, \Gamma \Rightarrow \gamma}{[\hat{d}]!\alpha, \Gamma \Rightarrow \gamma}\ (!\text{co}) \qquad \frac{[\hat{d}][b][c]\alpha, \Gamma \Rightarrow \gamma}{[\hat{d}][b\ ;\ c]\alpha, \Gamma \Rightarrow \gamma}\ (;\text{left}) \qquad \frac{\Gamma \Rightarrow [\hat{d}][b][c]\alpha}{\Gamma \Rightarrow [\hat{d}][b\ ;\ c]\alpha}\ (;\text{right}).$$

Note that Girard's intuitionistic linear logic ILL is a subsystem of SLL. The sequents of the form $[\hat{d}]\alpha \Rightarrow [\hat{d}]\alpha$ for any formula α are provable in cut-free SLL.

We now define a *sequence-indexed phase semantics* for SLL. The difference between such a semantics and the original phase semantics for ILL by Girard [1] is the definition of the valuations: whereas the original semantics has a valuation v, our semantics has an infinite number of sequence-indexed valuations $v^{\hat{d}}$ ($\hat{d} \in$ SE), where $v^{\emptyset}$ just works as v.

Definition 3. *An* intuitionistic phase space *is a structure* $\langle \mathbf{M}, cl, \mathrm{I} \rangle$ *satisfying the following conditions:*

1. $\mathbf{M} := \langle M, \cdot, 1 \rangle$ *is a commutative monoid with the identity* 1,
2. cl *is an operation on the powerset* $P(M)$ *of* M *such that, for any* $X, Y \in P(M)$,
   C1: $X \subseteq cl(X)$,
   C2: $clcl(X) \subseteq cl(X)$,
   C3: $X \subseteq Y$ *implies* $cl(X) \subseteq cl(Y)$,
   C4: $cl(X) \circ cl(Y) \subseteq cl(X \circ Y)$
   where the operation $\circ$ *is defined as* $X \circ Y := \{x \cdot y \mid x \in X \text{ and } y \in Y\}$ *for any* $X, Y \in P(M)$ *(the operation* cl *is called here* closure operation*),*
3. I *is a submonoid of* M *such that* $cl\{x\} \subseteq cl\{x \cdot x\}$ *for any* $x \in \mathrm{I}$.

Definition 4. *Constants and operations on* $P(M)$ *are defined as follows: for any* $X, Y \in P(M)$,

1. $\dot{\mathbf{1}} := cl\{1\}$,
2. $\dot{\top} := M$,
3. $\dot{\bot} := cl(\emptyset)$,

4. $X \dot{\rightarrow} Y := \{y \mid \forall x \in X\ (x \cdot y \in Y)\}$,
5. $X \dot{\wedge} Y := X \cap Y$,
6. $X \dot{\vee} Y := cl(X \cup Y)$,
7. $X \dot{*} Y := cl(X \circ Y)$,
8. $\dot{!}X := cl(X \cap \mathrm{I} \cap \dot{\mathbf{i}})$.

We define $D := \{X \in P(M) \mid X = cl(X)\}$. *Then,* $\mathbf{D} := \langle D, \dot{\rightarrow}, \dot{*}, \dot{\wedge}, \dot{\vee}, \dot{!}, \dot{\mathbf{i}}, \dot{\top}, \dot{\bot}\rangle$ *is called an* intuitionistic phase structure.

Definition 5. Sequence-indexed valuations $v^{\hat{d}}$ *for all* $\hat{d} \in$ SE *on an intuitionistic phase structure* $\mathbf{D} := \langle D, \dot{\rightarrow}, \dot{*}, \dot{\wedge}, \dot{\vee}, \dot{!}, \dot{\mathbf{i}}, \dot{\top}, \dot{\bot}\rangle$ *are mappings from the set of all propositional variables to* D. *Then,* $v^{\hat{d}}$ *for all* $\hat{d} \in$ SE *are extended to mappings from the set* Φ *of all formulas to* D *by:*

1. $v^{\hat{d}}(\mathbf{1}) := \dot{\mathbf{i}}$,
2. $v^{\hat{d}}(\top) := \dot{\top}$,
3. $v^{\hat{d}}(\bot) := \dot{\bot}$,
4. $v^{\hat{d}}(\alpha \wedge \beta) := v^{\hat{d}}(\alpha) \dot{\wedge} v^{\hat{d}}(\beta)$,
5. $v^{\hat{d}}(\alpha \vee \beta) := v^{\hat{d}}(\alpha) \dot{\vee} v^{\hat{d}}(\beta)$,
6. $v^{\hat{d}}(\alpha * \beta) := v^{\hat{d}}(\alpha) \dot{*} v^{\hat{d}}(\beta)$,
7. $v^{\hat{d}}(\alpha{\rightarrow}\beta) := v^{\hat{d}}(\alpha) \dot{\rightarrow} v^{\hat{d}}(\beta)$,
8. $v^{\hat{d}}(!\alpha) := \dot{!}v^{\hat{d}}(\alpha)$,
9. $v^{\hat{d}}([e]\alpha) := v^{\hat{d}\ ;\ e}(\alpha)$ *for any atomic sequence* e,
10. $v^{\hat{d}}([b\ ;\ c]\alpha) := v^{\hat{d}}([b][c]\alpha)$.

Remark that $v^{\hat{d}}([c]\alpha) := v^{\hat{d}\ ;\ c}(\alpha)$ holds for any $c, \hat{d} \in$ SE.

Definition 6. *An* intuitionistic sequence-indexed phase model *is a structure* $\langle \mathbf{D}, \{v^{\hat{d}}\}_{\hat{d}\in \mathrm{SE}}\rangle$ *such that* $\mathbf{D}$ *is an intuitionistic phase structure, and* $\{v^{\hat{d}}\}_{\hat{d}\in \mathrm{SE}}$ *is a set of sequence-indexed valuations. A formula* α *is* true *in an intuitionistic sequence-indexed phase model* $\langle \mathbf{D}, \{v^{\hat{d}}\}_{\hat{d}\in \mathrm{SE}}\rangle$ *if* $\dot{\mathbf{i}} \subseteq v^{\emptyset}(\alpha)$ *(or equivalently* $1 \in v^{\emptyset}(\alpha)$*) holds, and* valid *in an intuitionistic phase structure* $\mathbf{D}$ *if it is true for any sequence-indexed valuations* $\{v^{\hat{d}}\}_{\hat{d}\in \mathrm{SE}}$ *on the intuitionistic phase structure. A sequent* $\alpha_1, \cdots, \alpha_n \Rightarrow \beta$ *(or* $\Rightarrow \beta$*) is* true *in an intuitionistic sequence-indexed phase model* $\langle \mathbf{D}, \{v^{\hat{d}}\}_{\hat{d}\in \mathrm{SE}}\rangle$ *if the formula* $\alpha_1 * \cdots * \alpha_n{\rightarrow}\beta$ *(or* β*) is true in it, and valid in an intuitionistic phase structure if so is* $\alpha_1 * \cdots * \alpha_n{\rightarrow}\beta$ *(or* β*).*

Theorem 7 (Soundness). *If a sequent* S *is provable in* SLL*, then* S *is valid for any intuitionistic phase structures.*

3 Main Theorems

An expression $[\Gamma]$ represents the multiset consisting of all elements of a sequence (or multiset) Γ of formulas.

Definition 8. *We define a commutative monoid* $\langle M, \cdot, 1\rangle$ *as follows:*

1. $M := \{[\Gamma] \mid [\Gamma]$ *is a finite multiset of formulas*$\}$,
2. $[\Gamma] \cdot [\Delta] := [\Gamma, \Delta]$ *(the multiset union),*
3. $1 := [\,]$ *(the empty multiset).*

We define the following: for any $\hat{d} \in \mathrm{SE}$ *and any formula* α,

$$\|\alpha\|^{\hat{d}} := \{[\Gamma] \mid \vdash_{cf} \Gamma \Rightarrow [\hat{d}]\alpha\}$$

where $\vdash_{cf}$ *means "provable in cut-free* SLL*".*

Definition 9. *We define* $D := \{X \mid X = \bigcap_{i\in I} \|\alpha_i\|^{\emptyset}\}$ *for an arbitrary (non-empty) indexing set* I *and an arbitrary formula* α_i. *Then we define* $cl(X) := \bigcap\{Y \in D \mid X \subseteq Y\}$. *We define the following constants and operations on* $P(M)$*: for any* $X, Y \in P(M)$,

1. $\dot{\mathbf{1}} := cl\{1\}$,
2. $\dot{\top} := M$,
3. $\dot{\bot} := cl(\emptyset)$,
4. $X \mathrel{\dot{\to}} Y := \{[\Delta] \mid \forall\, [\Gamma] \in X\ ([\Gamma, \Delta] \in Y)\}$,
5. $X \mathbin{\dot{\wedge}} Y := X \cap Y$,
6. $X \mathbin{\dot{\vee}} Y := cl(X \cup Y)$,
7. $X \mathbin{\dot{*}} Y := cl(X \circ Y)$ *where* $X \circ Y := \{[\Gamma, \Delta] \mid [\Gamma] \in X$ *and* $[\Delta] \in Y\}$,
8. $\dot{!}X := cl(X \cap \mathrm{I} \cap \mathbf{1})$ *where* $\mathrm{I} := \{[[\hat{d}_1]!\gamma_1, ..., [\hat{d}_k]!\gamma_k] \mid [\hat{d}_1]!\gamma_1,\ ...,\ [\hat{d}_k]!\gamma_k$*: formulas*$\}$.

Sequence-indexed valuations $v^{\hat{d}}$ *for all* $\hat{d} \in \mathrm{SE}$ *are mappings from the set of all propositional variables to* D *such that* $v^{\hat{d}}(p) := \|p\|^{\hat{d}}$.

Proposition 10. *The structure* $\mathbf{D} := \langle D, \dot{\to}, \dot{*}, \dot{\wedge}, \dot{\vee}, \dot{!}, \dot{\mathbf{1}}, \dot{\top}, \dot{\bot}\rangle$ *defined above forms an intuitionistic phase structure.*

Lemma 11. *For any* $\hat{d} \in \mathrm{SE}$ *and any formula* α, $[[\hat{d}]\alpha] \in v^{\hat{d}}(\alpha) \subseteq \|\alpha\|^{\hat{d}}$.

Theorem 12 (Strong completeness). *If a sequent* S *is valid for any intuitionistic phase structures, then* S *is provable in cut-free* SLL.

Proof. If a sequent S is true, then the corresponding formula α is also true by Definition 6. If the formula α is true, then $[\,] \in v^{\emptyset}(\alpha)$ by Lemma 11. On the other hand, also by Lemma 11, $v^{\hat{d}}(\alpha) \subseteq \|\alpha\|^{\hat{d}}$ for any $\hat{d} \in \mathrm{SE}$, and hence $[\,] \in \|\alpha\|^{\emptyset}$, which means "$\Rightarrow \alpha$ is provable in cut-free SLL". This means that S is provable in cut-free SLL. **Q.E.D.**

Theorem 13 (Cut-elimination). *The rule* (cut) *is admissible in cut-free* SLL.

Proof. If a sequent S is provable in SLL, then S is valid by Theorem 7 (Soundness). By Theorem 12 (Strong completeness), S is provable in cut-free SLL. **Q.E.D.**

Theorem 14 (Undecidability). SLL *is undecidable.*

Proof. (Sketch): It is known that ILL is undecidable [3]. The proof of the undecidability of ILL is carried out by encoding Minsky machine. By Theorem 13, SLL is a conservative extension of ILL, and hence SLL can encode Minsky machine in the same way as in ILL. Therefore SLL is undecidable. **Q.E.D.**

Definition 15. SLL^- *is obtained from* SLL *by deleting* {(!left), (!right), (!co), (!we)}.

Definition 16. ILL^- *is obtained from* SLL^- *by deleting* {(;left), (;right)} *and all the expressions* $[\hat{d}]$ *appearing in the initial sequents and the logical inference rules. The names of the logical inference rules of* ILL^- *are denoted by labeling "*$\star$*" in superscript position, e.g.,* $(\rightarrow\mathrm{left}^\star)$.

Definition 17. *We fix a countable set* Ψ *of propositional variables, and define the sets* $\Psi_{\hat{d}} := \{p_{\hat{d}} \mid p \in \Psi\}$ $(\hat{d} \in \mathrm{SE})$ *of propositional variables where* $p_\emptyset := p$, *i.e.,* $\Psi_\emptyset := \Psi$. *The language (or the set of formulas)* $\mathcal{L}_s$ *of* SLL^- *is obtained from* Ψ, $\mathbf{1}, \top, \bot, \rightarrow, \wedge, *, \vee$ *and* $[b]$. *The language (or the set of formulas)* $\mathcal{L}$ *of* ILL^- *is obtained from* $\bigcup_{\hat{d}\in\mathrm{SE}} \Psi_{\hat{d}}$, $\mathbf{1}, \top, \bot, \rightarrow, \wedge, *$ *and* $\vee$.

A mapping f *from* $\mathcal{L}_s$ *to* $\mathcal{L}$ *is defined by:*

1. *for any* $p \in \Psi$, $f([\hat{d}]p) := p_{\hat{d}} \in \Psi_{\hat{d}}$,
2. $f([\hat{d}]\sharp) := \sharp$ *where* $\sharp \in \{\mathbf{1}, \top, \bot\}$,
3. $f([\hat{d}](\alpha \sharp \beta)) := f([\hat{d}]\alpha) \sharp f([\hat{d}]\beta)$ *where* $\sharp \in \{\rightarrow, \wedge, *, \vee\}$,
4. $f([\hat{d}][b\ ;\ c]\alpha) := f([\hat{d}][b][c]\alpha)$.

Let Γ be a set of formulas in $\mathcal{L}_s$. Then, an expression $f(\Gamma)$ means the result of replacing every occurrence of a formula α in Γ by an occurrence of $f(\alpha)$.

Theorem 18 (Embedding). *Let* Γ *be a multiset of formulas in* $\mathcal{L}_s$, γ *be a formula in* $\mathcal{L}_s$, *and* f *be the mapping defined in Definition 17. Then:* $\mathrm{SLL}^- \vdash \Gamma \Rightarrow \gamma$ *iff* $\mathrm{ILL}^- \vdash f(\Gamma) \Rightarrow f(\gamma)$.

Theorem 19 (Decidability). SLL^- *is PSPACE-complete.*

Proof. The provability of SLL^- can be transformed into that of ILL^- by Theorem 18. Since ILL^- is decidable, SLL^- is also decidable. Moreover, f in Definition 17 is a polynomial-time reduction. Since ILL^- is PSPACE-complete [3], SLL^- is also PSPACE-complete. **Q.E.D.**

4 Applications

We now consider an example of specifying secure password authentications on a network using SLL formulas. Consider a network that consists of three clients and a server. In order to obtain access to the server system, each client must try to login to the system in less than three attempts. As a password-protected

system, if a client inputs an incorrect user ID or password more than twice, then the server prevents the client from logging in. On the basis of a set of sequents in SLL that express the specification of secure password authentications and the behaviors of clients and servers, SLL can determine whether or not a client has successfully logged in to the server system.

Let c be one of three clients $c1$, $c2$, and $c3$; s be a server; and t be a counter from 0 to the maximal number k. By using the sequence modal operators, the SLL formula $[c; s; t]\alpha$ indicates that client c has sent a message α to server s at the t-th login attempt. In addition, the SLL formula $[s; c]\alpha$ indicates that server s returns a message α to client c. Accordingly, the sequence modal operators $[c; s]$ and $[s; c]$ represent the orders of message flows between client c and server s, and the sequence modal operator $[c; s; t]$ expresses the history of the client's messages on the basis of counter t.

The specifications of secure password authentications are described using SLL fomulas as follows:

$$
\begin{array}{l}
[c; s; t](\mathit{userID} * \mathit{password}), [s; c](\overbrace{\mathit{error} * \cdots * \mathit{error}}^{i}) \Rightarrow [s; c]\mathit{accept} \quad \text{for } i < 3 \\
[c; s; t](\mathit{incorrectUserID} * \mathit{password}) \Rightarrow [s; c]\mathit{error} \\
[c; s; t](\mathit{userID} * \mathit{incorrectPassword}) \Rightarrow [s; c]\mathit{error} \\
[c; s; t](\mathit{incorrectUserID} * \mathit{incorrectPassword}) \Rightarrow [s; c]\mathit{error} \\
[s; c](\mathit{error} * \mathit{error} * (!\mathit{error})) \Rightarrow [s; c]\mathit{reject}
\end{array}
$$

where $c \in \{c1, c2, c3\}$ and $t \in \{1, \ldots, k\}$. The fusion '$*$' and the exponential '!' in SLL are used to represent resource-sensitive aspects of errors arising from server s to client c. In the above sequents, the SLL formula $\mathit{error} * \mathit{error} * (!\mathit{error})$ implies that errors arise more than twice.

We describe two cases of attempts made by clients to login to the server system. Figure 1 shows examples of message flows between two clients $c1$ and $c2$ and server s. On the left hand side of the figure, two clients $c1$ and $c2$ send pairs of user ID and password to the server system s. The message flows are described as follows.

$$
\begin{array}{l}
[c1; s; 1](\mathit{incorrectUserID} * \mathit{password}) \\
[c2; s; 2](\mathit{userID} * \mathit{incorrectPassword}) \\
[c1; s; 3](\mathit{incorrectUserID} * \mathit{incorrectPassword}) \\
[c1; s; 4](\mathit{userID} * \mathit{password})
\end{array}
$$

On the basis of the above mentioned specifications and assumptions, SLL can prove the formula $[s; c1]\mathit{accept}$ as shown in the following proof-figure. We abbreviate userID, $\mathit{incorrectUserID}$, $\mathit{password}$, $\mathit{incorrectPassword}$, error, accept, and reject to u, iu, p, ip, e, a, and r, respectively.

$$
\dfrac{\Rightarrow [c1; s; 4](u * p) \qquad \dfrac{\dfrac{\vdots\ P_1 \qquad \vdots\ P_2}{\Rightarrow [s; c1]e \quad \Rightarrow [s; c1]e} \qquad S}{\Rightarrow [s; c1](e * e) \qquad [c1; s; 4](u * p) \Rightarrow [s; c1]a}}{\Rightarrow [s; c1]a}
$$

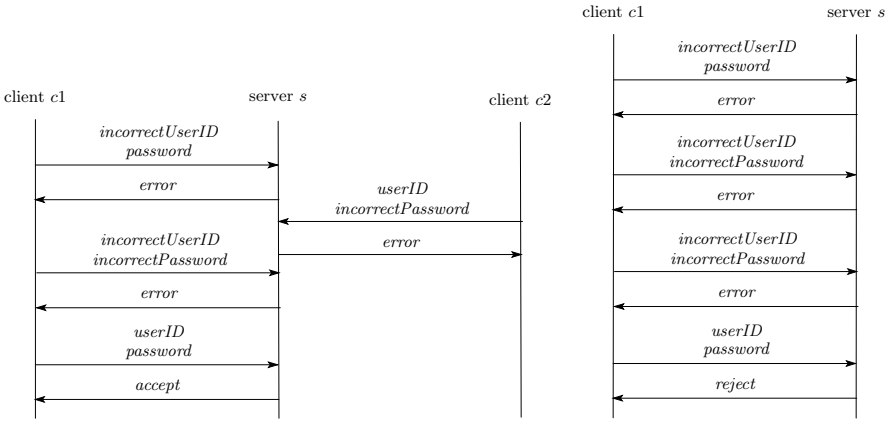


Fig. 1. Message flow during successful and failed logins

where S, P_1 and P_2 are respectively:

$$\dfrac{\dfrac{[c1;s;4](u*p),[s;c1](e*e)\Rightarrow[s;c1]a,}{\Rightarrow[c1;s;1](iu*p)\quad[c1;s;1](iu*p)\Rightarrow[s;c1]e}\Rightarrow[s;c1]e}{\Rightarrow[c1;s;3](iu*ip)\quad[c1;s;3](iu*p)\Rightarrow[s;c1]e}\Rightarrow[s;c1]e}\ .$$

On the right hand side of the figure, client $c1$ inputs the user ID and password to server s four times. Only the last pair of user ID and password is correct. In this scenario, the message flows between client $c1$ and server s are described as follows.

$[c1;s;1](incorrectUserID * password)$
$[c1;s;2](incorrectUserID * incorrectPassword)$
$[c1;s;3](incorrectUserID * incorrectPassword)$
$[c1;s;4](userID * password)$

On the basis of the above mentioned specifications and assumptions, SLL can prove the formula $[s;c1]reject$ as shown in the following proof-figure.

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots\ P_1}{\Rightarrow[s;c1]e}\quad\dfrac{\vdots\ P_2}{\Rightarrow[s;c1]e}}{\Rightarrow[s;c1](e*e)}\quad\dfrac{\vdots\ P_3}{\Rightarrow[s;c1]e}}{\Rightarrow[s;c1](e*e*(!e))}\quad\dfrac{}{[s;c1](e*e*(!e))\Rightarrow[s;c1]r}}{\Rightarrow[s;c1]r}$$

where P_1, P_2 and P_3 are respectively:

$$\dfrac{\Rightarrow[c1;s;1](iu*p)\quad[c1;s;1](iu*p)\Rightarrow[s;c1]e}{\Rightarrow[s;c1]e}$$

$$\dfrac{\dfrac{\Rightarrow [c1;s;2](iu * ip) \quad [c1;s;2](iu * ip) \Rightarrow [s;c1]e}{\Rightarrow [s;c1]e} \quad \dfrac{\Rightarrow [c1;s;3](iu * ip) \quad [c1;s;3](iu * ip) \Rightarrow [s;c1]e}{\Rightarrow [s;c1]e}}{\Rightarrow [s;c1]e}\ .$$

SLL can represent the above specification of secure password authentications, but it is undecidable. In order to avoid the undecidability, we use the decidable fragment SLL$^-$ to describe the modified specification that is obtained from the specification based on SLL by replacing the sequent $[s;c](error * error * (!error)) \Rightarrow [s;c]reject$ with $[s;c](\overbrace{error * \cdots * error}^{j}) \Rightarrow [s;c]reject$ for $3 \leq j \leq k$. The expression $error * error * (!error)$ in SLL is replaced with $\overbrace{error * \cdots * error}^{j}$ in SLL$^-$. This is redundant when compared with the SLL formulas because $k-2$ formulas are added instead of the fusion connective. On the basis of the above mentioned specifications and assumptions, SLL$^-$ can prove the formula $[s;c1]reject$ as shown in the following proof-figure.

$$\dfrac{\dfrac{\dfrac{\vdots\ P_1}{\Rightarrow [s;c1]e} \quad \dfrac{\dfrac{\vdots\ P_2}{\Rightarrow [s;c1]e} \quad \dfrac{\vdots\ P_3}{\Rightarrow [s;c1]e}}{\Rightarrow [s;c1](e * e)}}{\Rightarrow [s;c1](e * e * e)} \qquad [s;c1](e * e * e) \Rightarrow [s;c1]r}{\Rightarrow [s;c1]r}$$

In SLL$^-$, $[s;c1]reject$ cannot be proven if more than k errors arise because SLL$^-$ does not have the expressive power of SLL.

Acknowledgments. We are partially supported by the Japanese Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Young Scientists (B) 20700015 and (B) 20700147.

References

1. Girard, J.-Y.: Linear logic. Theoretical Computer Science 50, 1–102 (1987)
2. Kamide, N., Kaneiwa, K.: Extended full computation-tree logic with sequence modal operator: Representing hierarchical tree structures. In: Nicholson, A., Li, X. (eds.) AI 2009. LNCS (LNAI), vol. 5866, pp. 485–494. Springer, Heidelberg (2009)
3. Lincoln, P., Mitchell, L., Scedrov, A., Shankar, N.: Decision problems for propositional linear logic. Annals of Pure and Applied Logic 56, 239–311 (1992)
4. Miller, D.: An overview of linear logic programming. London Mathematical Society Lecture Note Series (Linear logic in computer science), vol. 316, pp. 119–150. Cambridge University Press, Cambridge (2004)
5. Okada, M.: A uniform semantic proof for cut-elimination and completeness of various first and higher order logics. Theoretical Computer Science 281, 471–498 (2002)
6. Pym, D.J., O'Hearn, P.W., Yang, H.: Possible worlds and resources: The semantics of BI. Theoretical Computer Science 315, 257–305 (2004)
7. Russell, S., Norvig, P.: Artificial intelligence: A modern approach, 2nd edn. Pearson Education, Inc., London (2003)
8. Wansing, H.: The Logic of Information Structures. LNCS (LNAI), vol. 681, 163 pages. Springer, Heidelberg (1993)

Frame-Based Dialogue Management Automated Error Recovery Approach

Tomáš Nestorovič

University of West Bohemia in Pilsen, Univerzitní 8,
30614 Pilsen, Czech Republic
nestorov@kiv.zcu.cz

Abstract. This paper focuses on our recent work in frame-based dialogue management. Despite frames are frequently used in commercial speech applications, they lack enough automation for error recovery within a dialogue. In this paper we present and demonstrate our mechanism that fills this gap by automatically tracking and managing the dialogue flow causality.

Keywords: Dialogue management; dialogue systems; frame-based dialogue management; frame structure extension, management automation.

1 Introduction

The dialogue management is one of artificial intelligence disciplines that concerns human-computer interaction (HCI). Its aim is to find machine's best response, given user's (spoken) input and interaction history. During the past decades, many approaches emerged. What they have in common is the aim to manage and elicit knowledge within a dialogue, however, their theoretical backgrounds differ. Ranging from simple finite state machines to intelligent agents, and Markov decision networks, there is a wide collection of methods on how to implement a dialogue manager.

We follow the way of frames whose potential lies in commercial spoken dialogue applications [1, 2]. We find here a lack of automation for error recovery within a frame-driven dialogue: if a user corrects the system, then s/he in fact reopens a past dialogue topic (e.g., a departure city name topic is reopened with user's utterance "No, I said from the Hague" after the system responded "The next train from Haarlem to Amsterdam leaves at 9 a.m."). Certainly, one of applicable solutions might be to watch for updates of slot values and trigger proper reactions within OnFilled-like event handlers. However, this approach has two drawbacks: 1) (implementation-related) once the logics gets more complicated, it will be hard to keep track of where to "jump" next within a frame, and 2) (theoretical-related) this approach is well applicable for flat frames only. As a response, we have developed a journaling system as an extension to the common construction of frames, that enables us: 1) to detect no longer valid information after the user makes changes within a dialogue, and 2) to "rollback" nested frames structure in accordance with the changes made, hence, putting the structure into a causally consistent state.

In the rest of the paper, we explain our approach (Section 2), and provide an example (Section 3). Finally, we suggest future work and conclude (Sections 4 and 5).

J. Li (Ed.): AI 2010, LNAI 6464, pp. 32–41, 2010.

2 Journaling System and Causality Tracking Mechanism

Above, we introduced the error recovery problem in dialogue management. To define it formally, we consider an *error* to be a change of a piece of information that has already been used by the system (thus we do not perceive an unexploited information change to be an error). When attempting to recover from an error, the system must throw the replaced piece of information away (along with all depending pieces) and return with the interaction to a specific point in the dialogue (i.e., "jump" back), reopening one of the past topics.

As a solution to the error recovery problem, we have developed the journaling system and the causality tracking mechanism, both of which are accommodated in our dialogue manager basic architecture [3] that operates above a nested frames structure. Before we proceed, let us make three assumptions about the frames environment.

- We consider the notion of frames to be rather "concept-like", i.e., we expect that each frame is allowed to carry a single domain information at a time (e.g. city name information). Having a number of simple concept-like frames means to be able to catch the soft notion of information about a domain and to enable sophisticated manipulation with it [4]. Presumably, dialogue management specific operations (disambiguation, validation, etc.) get simpler this way, as the concept-like nested structure better fits entities mentioned in the dialogue.
- Additionally, we assume each frame is equipped with a means to provide instructions on "what to do next." In our case, the means is a *message queue* whose content is filled by both the form interpretation mechanism (see [3]) and causality tracking mechanism (see below). The messages are like "interpret slot S" or "query the user for specific information for slot S."
- The form interpretation mechanism must be *state-less*, i.e., it must not make any presumption about the current state of the frames structure. However, this does not imply that it is supposed to be a blind routine.[1] In fact with this demand, we want the mechanism to be open to further extensions. More specifically the causality tracking mechanism is intended to override the interpretation mechanism's behaviour in cases of recovering from an error.

2.1 Journaling System

With having basic nested frame structure,[2] we extended each frame with a *journal* to track its activity during a dialogue. The activity involves *frame slot interpretation*, *information reading*, and *information writing*. Each of the activities occupies a single *record* in the journal. For the purpose of simplicity in the remainder of the paper, we will use the terms I-record, R-record, and W-record to respectively refer to these activities. Additionally, the journal disposes of a *rollback function* that gives it a possibility to *temporarily* forget recent activities in the reversed order they were made

[1] In our case, it does an optimization regarding what piece of information to validate – if the confidence score is high, the interpretation mechanism does not instruct the frame to initiate a validation subdialogue to confirm the correctness.

[2] The frames do not need to have any extra features to support our approach.

(temporarily = particular records are moved to a *REDO* part of the journal, waiting to be recovered eventually – see the causality tracking mechanism below).

Let us have a look at the records and their parameters. The I-records acquire a single parameter – the slot that has been interpreted. Denoting the slot as *S*, we shortly can express an I-record as *I(S)*. Obviously, the sequence of I-records in a journal indicates the trace of the frame interpretation. The R-records acquire three parameters – the *Reader* parameter refers to a frame that being at slot *S* initiated the reading of slot *T*. Hence, the formal expression of the R-record takes the form *R(Reader,S,T)*. Finally, the W-records acquire two parameters – the slot *S* and its corresponding value *V*, resulting in expression *W(S,V)*.

One of our aims with the error recovery has been to reveal where to "jump" next in an automated manner (i.e. potentially reopen a closed topic). Having a *single* frame in which a *single* error on slot *E* emerged means the consecutive records *R(Reader,S,E)* and *W(E,V)* are part of its journal. Apparently, the reading *R(Reader,S,E)* is no longer valid, thus the journal needs to be rolled back until the reading disappears from it. The last rolled-back I-record *I(S)* indicates the new reinterpretation state for this frame and the start of the potentially reopened topic (see below).

Above, we concerned the single frame case. A structure of frames in which each contains a self-contained journal implies the *journaling system*.

In the journaling system, the W-records play a crucial role. Not only they can trigger the rollback process, but they also indicate that a frame has been bound with a subframe (in cases where the subframe is the allowed value). The question that arises now is: How to deal with these subframes, once we need to roll back their corresponding W-records? Let us consider the dialogue context in Fig. 1 and suppose the *W(Train,Arrival)* record is rolled back from the Train journal. There are three solutions how to deal with the Arrival frame (generally applicable to any other frames combination): 1) keeping it, 2) recurrently rolling back the entire Arrival branch, or 3) rolling back just immediate relation between the frames (shadowed in Fig. 1). The second solution was used in our previous version of the journaling system [3], however, we dropped it due to being too restrictive for our future work (Section 4).

Our implementation accounts for the third solution since it makes just minor interference with the context, instead of rapidly changing it. If a frame is being rolled back, only its immediate relations are moved to the REDO journal – all subframes stay in the context. Obviously, this is a trade-off: from the dialogue context point of

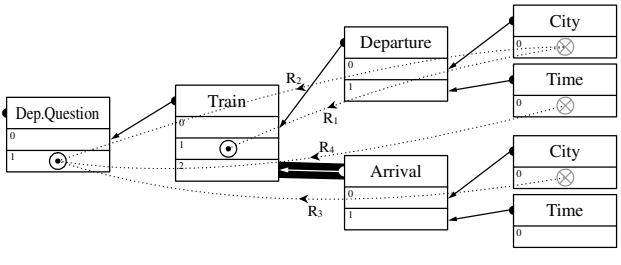


Fig. 1. Information readings ($R_{[1,2,3,4]}$) realized within the given context instance. *Left sided numbers* in serve as slot identifiers. *Shadowed* is a rolled-back relation.

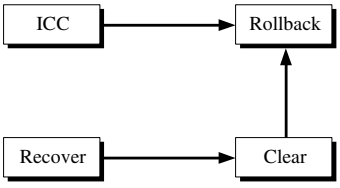


Fig. 2. Elemental procedures interaction chart with edges denoting the "triggers" relationship

view, the subframes seem like they were not bound, while from the user's point of view, they do not exhibit a need to be generalized.[3] This apparent ambiguity exactly matches the uncertainty about the rolled back data: no sooner than after the REDO journal is cleared, the system is forced to pose a generalization question.

The journaling system provides us with a means to manage an extra information about the context (journals with records), however, it is unable to make any decisions. The executive part of our approach is the causality tracking mechanism.

2.2 Causality Tracking Mechanism

The causality tracking mechanism consists of four algorithms that govern the journaling system functionality. These algorithms regard the journal *initial content creation*, *rollback*, *recovery*, and *clean up*. For the purpose of comfortably referring to each one of them, let us introduce the terms *ICC*, *Rollback*, *Recover*, and *Clear*, respectively. Fig. 2 and the top-level algorithm in Fig. 3 give an overview of the entire process and interaction.

```
Procedure PerformInterpretationOfSlot (S) {
   1. Let F denote a frame with a journal J.
   2. While slot S ∈ F is interpreted {
      2.1. Recover(S) – attempt to recover formerly rolled-back records from J_REDO.
      2.2. Interpret(S) – perform the common interpretation of slot S.
      2.3. ICC(S) – based on the interpretation success, push new records into J.
     }
}
```

Fig. 3. Interpretation algorithm of a journaled frame

Before engaging in formal descriptions, let us note that for the algorithms to not introduce unnecessary variables, we will adhere to the Prolog-like underscore notation ("_") to mark that a given variable is not significant within a particular portion of algorithm – e.g., *R(Reader,_,T)* denotes the R-record whose reader's interpretation state is insignificant.

Let us naturally start with the *ICC algorithm* (Fig. 4). It defines how to deal with records produced during the frame interpretation cycle (step 2.2 in Fig. 3).

[3] The term "generalization" refers to providing a frame with a super-frame, e.g. in a time-table domain, we can generalize the Time with either the Departure or Arrival frames (Fig. 8).

Additionally, it triggers the rollback process if an error is encountered within the given frame (step 2.3.1 in Fig. 4).

The Rollback algorithm (Fig. 5) shows that rolling back the I-records and W-records is a straight-forward manipulation – they are simply moved to the REDO journal without any further processing (steps 3.1.2 and 3.1.5 in Fig. 5). However with the R-records, only those for which a given frame was a reader are considered (step 3.1.3), otherwise they are skipped (step 3.1.4). The reason is that all frames take responsibility over their own actions only (i.e., their own behaviour within the nested environment), and hence, leave traces of external interaction unattended. As a result after the rollback process, each reader's interpretation state equals the state just prior

```
Procedure ICC (S) {
  1. Let F denote a frame with a journal J. Let S and T denote slots.
  2. While interpreting slot S ∈ F produces a record D {
     2.1. D = I(S)  ⇒  add D into J
     2.2. D = R(F,_,T)  ⇒  add D into J ∧ add D into journal of a frame that owns T
     2.3. D = W(T,_)  ⇒
        2.3.1. R(_,_,T) ∈ J  ⇒  Rollback(T)
        2.3.2. combine W-record's value with T's current value
        2.3.3. add D into J
     }
}
```

Fig. 4. Journal initial content creation algorithm

```
Procedure Rollback (E0) {
  1. Let E0 denote the initial set of slots whose values have changed.
     Let S, T and U be slots. Let N denote a dialogue context relation.
  2. Find the set E under partial order of E0 (i.e., let E contain all slots
     transitively dependent on E0, and it holds E ∩ E0 = E0).
  3. For each slot T ∈ E, owned by a frame F equipped with journal J {
     3.1. While ∃ R(F,_,T) ∈ J {
        3.1.1. Let D be the most recent record in J
        3.1.2. D = I(_)  ⇒  move D to J_REDO
        3.1.3. D = R(F,S,U)  ⇒
           3.1.3.1. U ∈ E  ⇒  let S define F's new interpretation state
                              ∧ exclude D from J
                              ∧ exclude D from journal of frame which U is part of
           3.1.3.2. else move D to J_REDO
           3.1.3.3. Go to 3.1
        3.1.4. D = R(_,_,_)  ⇒  skip processing D, i.e., keep D in J
        3.1.5. D = W(_,N)  ⇒  exclude N from the context ∧ move D to J_REDO
     }
  }
}
```

Fig. 5. Journal rollback algorithm

to any reading occurred (step 3.1). Note that rolling a frame journal back means also to remove relations to subframes of that frame (step 3.1.5), as discussed above.

To roll back a frame interpretation history gives that frame a possibility to recover it. The main point about the *Recover algorithm* (Fig. 6) is that it compares the oldest historical interpretation state (an I-record) with the current one (step 2.1.1 in Fig. 6). Their difference signs that the current interpretation has taken a distinct branch of the background logic, and the rest of the REDO journal can no longer be trusted, i.e. is to dispose (step 2.1.2). Note that relations which have been removed during the Rollback procedure are included back to the dialogue context (step 2.3).

The *Clear algorithm* concerns the cleanup of a REDO journal (Fig. 7). Since frames take responsibility over their own actions only, we do not need to check the origin of R-records explicitly – we know that all these readings were initiated by the frame that the given REDO journal is a part of.

```
Procedure Recover (S) {
  1. Let F denote a frame with a journal J.
     Let S and T denote slots, and N be dialogue context relation.
  2. While any record D ∈ J_REDO relates to slot S ∈ F {
     2.1. D = I(S)  ⇒
        2.1.1. D is the oldest I-record in J_REDO  ⇒
           2.1.1.1. ∀ R(F,S,_) are valid  ⇒  recover D to J
           2.1.1.2. else remove all R(F,S,_) ∧ reinterpret S
        2.1.2. else Clear(J_REDO)
     2.2. D = R(F,S,_)  ⇒  recover D to J
     2.3. D = W(S,N)  ⇒  recover D to J ∧ include N back into the context
                         ∧ combine W-record's value with S's current value
  }
}
```

Fig. 6. Journal recovery algorithm

```
Procedure Clear (J_REDO) {
  1. Let T denote a slot, and N denote a dialogue context relation.
  2. While J_REDO (owned by frame F) is not empty {
     2.1. Exclude a record D from J_REDO
     2.2. D = I(_)  ⇒  dispose D
     2.3. D = R(_,_,T)  ⇒  exclude D from journal of frame which T is a part of
                           ∧ dispose D
     2.4. D = W(_,N)  ⇒  let M be a set of interpretation states of
                                 readers engaged with relation N
                         ∧ Rollback(M)
                         ∧ acquire generalization of frame X that N bounded with F
                         ∧ dispose D ∧ dispose N
  }
}
```

Fig. 7. Journal cleanup algorithm

3 Example

We have developed a simplified[4] time-table domain application (Fig. 8) that provides departure and arrival times information. In the model, solid lines represent standard relations (to maintain relevant information connections), while the dotted ones refer to disambiguation relations (to express a detailed description of a frame – used when accessing a database). The numbers in the domain ("0…5") refer to acceptable amount of database query results to present to the user (the number of five was chosen according to an empirical rule of speech interface design stating that users are unable to memorize more than approximately seven last options within a list [5]).

Let us consider the dialogue snippet in Table 1, that results in the context fragment depicted in Fig. 1. At the end, the system utters a particular transportation means in Q/1 (Departure***Q***uestion frame, slot **1**), and an additional back-end reading is performed in T/1 (***T***rain). Each frame's journal content is depicted in Fig. 9, which also serves as a trace of the interpretation algorithm (I-records) as time is involved. In its final answer, the system makes three train suggestions, each internally converting into a semantics that in turn replaces the user's initial search pattern – thus the system substitutes generally specified times with particular departure and arrival times of particular transportation means. This is for the trains to appear in the dialogue history.

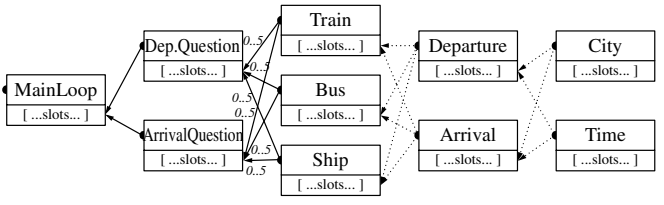


Fig. 8. Time-table domain model; *solid edges* represent standard relations, *dotted* ones refer to disambiguation relations

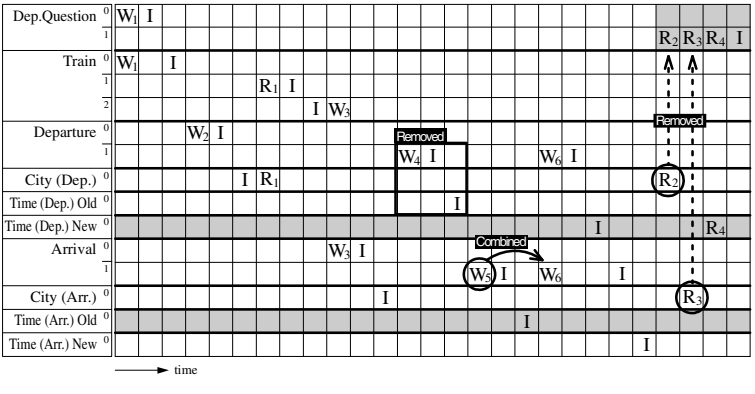


Fig. 9. Journals contents; *shadowed parts* are stored in the REDO part of journals. *Numbers* on the right of the frame names identify particular slots.

[4] The model does not consider changes and different kinds of trains.

Table 1. Dialogue snippet between an user (U) and the system (S); italicized comments relate to system's background actions

Agent	W-record	Agent's utterance
U	W_1	I need to get information about train departures.
S	W_2	*[The departure city is created and initialized to the current city (Haarlem) during the interpretation process.]*
	W_3	Train departure times. Where do you want to travel to?
U		Amsterdam.
S	W_4	Amsterdam. What time approximately would you like to depart?
U	W_5	I must get there before 5 p.m. *[The user disambiguates the train other way. The system's attempt is removed from the journals.]*
S	W_6	*[User's utterance is anchored, causing the number of database query results to pass the limit of 5 transportation means at most. The system presents them. Each one is transformed to a semantics and anchored within the context, replacing Departure.Time, and Arrival.Time concepts. Note that we currently do not make use of system utterance planning – employing ellipsis in the following utterance would be appropriate.]* The next train from Haarlem to Amsterdam leaves at 9 a.m. The next train from Haarlem to Amsterdam leaves at 11 a.m. The next train from Haarlem to Amsterdam leaves at 1 p.m.

Table 2. Dialogue snippet between an user (U) and the system (S)

Agent	W-record	Agent's utterance
U	W_7	No, I want to leave from the Hague.
S	W_8	The next train from the Hague to Amsterdam leaves at 10 a.m. The next train from the Hague to Amsterdam leaves at 12 a.m.

Furthermore, consider the user wants to make some additional changes regarding the train which s/he intends to find (i.e., introduces an error). Table 2 shows the corresponding dialogue snippet. With changing the City of departure, the DepartureQuestion and Train frames are rolled back (Fig. 10), however, the rolled-back fragments still remain stored in the REDO parts of the journals. The original City of departure is no longer bound to the Departure frame, and consequently, neither R_1 nor R_2 (Fig. 1) are valid. With rolling back the W_3-record, only the relation between Arrival and Train is moved to the Train's REDO journal, and the remainder of the Arrival branch stays unaffected (Fig. 1). To keep track of the context parts that were modified, notification messages with D/0, T/0 and Q/0 need to be sent to the Departure, Train, and DepartureQuestion, respectively. Upon this, all necessary steps to recover the context from the error have been made and the reinterpretation may begin. It starts with obtaining the messages, and thus, revisiting the City of departure. Next, it continues reevaluating T/1 and moves to T/2. Here, no subframe is found, however, the Train's REDO journal contains a relation to a subframe. Recovering it, the formerly cut off Arrival branch is recovered. Finally, the interpretation reaches Q/1 and a new prompt is generated – “The next train from the Hague to Amsterdam

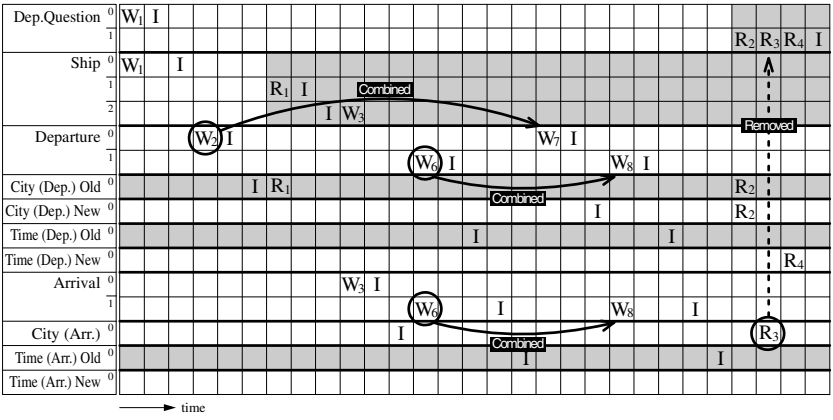


Fig. 10. Journals contents; *shadowed parts* are stored in the REDO part of journals. *Numbers* on the right of the frame names identify particular slots.

leaves at 10 a.m." Note that in neither cases the Departure frame is rolled back. The reason is that the conditions for triggering a rollback are not met – information in the frame is always only rewritten but never read.

4 Future Work

The journaling system has been applied in an simplified time-table domain to prove the feasibility of the idea and tune the approach. Apart of the time-table domain, we are going to apply the presented dialogue manager in a personal assistance domain, considering e-mails and appointments management. This domain shares a lot with the time-table (form-filling style of interaction and vastly passive data concepts). However, the distinction lies in the possibility of negotiation with the user (e.g., regarding two overlapping appointments), hence we expect minor changes to the dialogue manager, but no rapid changes to the error recovery approach.

5 Conclusion

The research goal we follow is to create a generic dialogue manager. Our approach utilizes frames technique for context knowledge representation. The mechanisms are based on known approaches we adjusted to fit our purposes. To summarize them, in [1] we found a motivation for nested frames technique, from [6] we adjusted the Form Interpretation Mechanism to work recursively using message passing – as a result, we treat frames as intentions, stacking them to keep track of focus in the dialogue, similarly as Grosz and Sidner do [7]. Finally, our disambiguation process is inspired by [8], however, we extended it to work in nested frames environment.

In this paper, we presented and demonstrated our approach to automated error recovery in dialogue management. The example above regarded just simple revision of the departure city which in fact could be handled in a simpler way *manually* (at the

system design/implementation stage). However, we aimed to show how to accomplish the same task by passing the responsibility for it to the *machine*, thus at first sight, the process may look like a redundant machinery. Nevertheless, considering it constitutes an uninformed methodology to solve the error recovery problem, we needed to sacrifice some of frame-based approach habits, and do some minor cuts to the way frames are commonly dealt.

To accommodate our solution in a dialogue manager, the following steps must be satisfied: 1) having common nested frames environment (common = frames do not need to be equipped with any special features, e.g. VoiceXML's [6] implementation of frames is sufficient, however, VoiceXML as such is insufficient, as it does not support nested frames), 3) extending the nested structure with the journaling system and a set of records (regarding the I-, R-, and W-records are mandatory), 3) implementing the four procedures handling journals initial content creation, rollback, recovery, and cleanup, 4) choosing the communication means between the causality tracking mechanism and the interpretation mechanism (in our case, the message passing is employed), and 5) making the causality tracking mechanism dominant to the interpretation mechanism when handling an error (by having the interpretation mechanism state-less, we can simply skip this final step).

Acknowledgement. The author would like to thank to the reviewers for their effort on improving the paper. This work was supported by grant no. 2C06009 Cot-Sewing.

References

1. Pieraccini, R., Suendermann, D., Dayanidhi, K., Liscombe, J.: Are We There Yet? Research in Commercial Spoken Dialog Systems. In: Matoušek, V., Mautner, P. (eds.) TSD 2009. LNCS, vol. 5729, pp. 3–13. Springer, Heidelberg (2009)
2. Gustafson, J.: Developing Multimodal Spoken Dialogue Systems – Empirical Studies of Spoken Human-Computer Interaction. KTH, Department of Speech, Music and Hearing, Stockholm (2002)
3. Nestorovič, T.: Towards Flexible Dialogue Management Using Frames. In: Matoušek, V., Mautner, P. (eds.) TSD 2009. LNCS, vol. 5729, pp. 419–426. Springer, Heidelberg (2009)
4. van Zanten, G.V.: User modelling in adaptive dialogue management. In: EUROSPEECH, pp. 1183–1186 (1999)
5. Gardner-Bonneau, D.: Guidelines for Speech-enabled IVR Application Design. In: Gardner-Bonneau, D. (ed.) Human Factors and Voice Interactive Systems, pp. 147–162. Kluwer Academic Publishers, Boston (1999)
6. W3C: Voice Extensible Markup Language, Version 2.0, `http://www.w3.org/TR/voicexml20/`
7. Grosz, B.J., Sidner, C.L.: Attention, Intention and the Structure of Discourse. Computational Linguistics 12, 175–204 (1986)
8. McGlashan, S.: Towards Multimodal Dialogue Management. In: Twente Workshop on Language Technology 11 (TWLT11), pp. 1–10 (1996)

Characterizing Relevant Belief Revision Operators

Laurent Perrussel[1], Jerusa Marchi[2], and Dongmo Zhang[3]

[1] IRIT - Université de Toulouse
Toulouse - France
laurent.perrussel@irit.fr
[2] Universidade Federal de Santa Catarina
Florianópolis - SC - Brazil
jerusa@das.ufsc.br
[3] University of Western Sydney
Sydney - Australia
dongmo@scm.uws.edu.au

Abstract. This paper introduces a framework for relevant belief revision. We represent agent's beliefs in prime implicants and express agent's preference on beliefs as a pre-order over terms. We define a belief revision operator via minimising the change of the prime implicants of the existing beliefs and the incoming information with respect to agent's preferences. We show that such a belief revision operator satisfies Katsuno and Mendelzon's postulates for belief revision as well as Parikh's postulate for relevant revision. This paper demonstrates a natural way to identify relevance of beliefs and an implementation of Parikh's relevant belief revision.

1 Introduction

Belief revision is the process of incorporating new pieces of information into a set of existing beliefs. It is usually assumed that the operation follows the following two principles: (i) the resulting belief set is consistent and (ii) the change on the original belief set is minimal. Several formalisms of belief revision have been proposed in the literature (see [8] for more details). The most influential work is the AGM paradigm which characterises the belief revision operation by a set of plausible axioms, generally referred to as the AGM postulates [1].

Despite of the popularity of the AGM paradigm, the AGM postulates are not sufficient to capture the notion of minimal change. As stressed by Parikh in [19], the full meet revision operator (removing all statements of the original beliefs and keeping only the new piece of information) satisfies the AGM postulates, which is obviously not a minimal change. In order to avoid counter-intuitive change of beliefs, Parikh proposed an additional postulate to the AGM postulates, which characterises the notion of relevant revision. A revision is said to be *relevant* if it enables to keep all the initial beliefs of an initial belief set ψ that are not related to the new piece of information after the belief revision operation. Formally speaking, if a statement of ψ does not use any propositional symbols that are used in the new piece of information μ, then this statement should belong to the resulting belief set. In other words, Parikh's postulate is grounded in the symbols used in ψ and μ. However, Parikh did not provide an actual belief revision operator that satisfies his postulate and the AGM postulates. The main difficulty

J. Li (Ed.): AI 2010, LNAI 6464, pp. 42–51, 2010.

is that it is hard to find simple and intuitive criteria that separate relevant information from irrelevant information. Makinson in [16] proposed a formal approach that is able to split logical symbols that are used in a belief set. However, no construction of belief revision operator was provided based on this language splitting approach. Peppas *et al.* in [20] proposed a model based construction of belief revision operator (based on systems of spheres) that implements the AGM postulates and Parikh's postulate. However, relevance by its nature is a syntactical issue. A model based approach at most provides a peripheral solution.

This paper aims to offer a syntactical construction of belief revision operator based on prime implicants. In [3], Bittencourt *et al.* proposed a syntax-based belief revision operator that is constructed by using prime implicants and prime implicates. Such a construction provides a natural way to identify relevance of beliefs. However, the construction is based on Dalal's distance, which cannot capture the notion of minimal change based on general preference orderings. In this paper we redefine the belief revision operator based on minimal change on general preference orderings. We define a belief revision operator via minimising the change of the prime implicants of the existing beliefs and the incoming information. We show that such a belief revision operator satisfies Katsuno and Mendelzon's postulates for belief revision as well as Parikh's postulate for relevant revision. Our approach provides a clear and simple way to address the belief relevance issue and a natural implementation of Parikh's relevant belief revision.

The paper is organised as follows. Section 2 reviews the notions of implicant and prime implicant. Section 3 defines a class of revision operators based on the prime implicant representation of beliefs. Section 4 shows that Parikh postulate holds for this class of revision operators. Finally, we conclude the work and discuss the related work with a perspectives of possible future work.

2 Preliminaries

Let $P = \{p_0, \ldots, p_n\}$ be a finite set of propositional symbols and $LIT = \{L_0, \ldots, L_{2n}\}$ be the set of the associated literals: $L_i = p_j$ or $\neg p_j$. Let $\overline{L}$ be the complementary literal, s.t. $\overline{L} = p$ (respectively $\neg p$) iff $L = \neg p$ (respectively p). Let $\mathcal{L}(P)$ be the propositional language associated to P and $\psi \in \mathcal{L}(P)$ be an ordinary formula. Let Lang be the function that assigns to each formula the set of propositional symbols that are contained in the formula, i.e., $\mathsf{Lang} : \mathcal{L}(P) \mapsto 2^P$.

Let $\mathcal{W}(P)$ be the whole set of propositional interpretations associated to P (for the sake of conciseness, hereafter we skip parameter P) and $\models$ the satisfiability relation. Let $[\![\psi]\!]$ be the set of propositional interpretations that satisfy ψ (the models of ψ).

Any formula can be represented in a disjunctive normal form (DNF). Given a formula ψ, let DNF_ψ be a DNF of ψ. Assume that $DNF_\psi = D_0 \vee \cdots \vee D_w$ be the *disjunction of terms*, where each term D_i is a *conjunction of literals*: $D_i = L_0 \wedge \cdots \wedge L_k$. Let $\overline{D}$ be the mirror of term D s.t. $\overline{D} = L_0 \wedge \cdots \wedge L_{k_D}$ iff $D = \overline{L_0} \wedge \cdots \wedge \overline{L_{k_D}}$.

A term D is an *implicant* of ψ if $D \models \psi$. A term D is said to be a *prime implicant* [21] of ψ if D is an implicant of ψ and for any term D' such that $D' \subseteq D$, we have $D' \not\models \psi$, i.e., a prime implicant of a formula ψ is an implicant of ψ without any subsumed terms. In the following, terms can be seen as sets of literals. Hereafter, we

frequently switch between the logical notation and the set notation. We write $D - D'$ to denote the subtraction operation over terms, that results from removing all literals that occur in D' from D, that is, $D - D' = \{L \in D : L \notin D'\}$. Although the definition of prime implicant includes contradictory terms, because a contradiction is an implicant of any formula, in the sequel we only consider terms that do not have any pair of contradictory literals. We also do not consider implicants with redundant literals (i.e. a literal can only appear at most once in an implicant). Let $\mathcal{D}_\psi$ be the set of all the non contradictory and non redundant implicants of ψ and $\mathcal{D}$ be the set of all non contradictory and non redundant terms. Notice that since P is finite, $\mathcal{D}$ is also finite.

We define PI_ψ as a disjunction of all non contradictory prime implicants of ψ such that $\psi \equiv PI_\psi$. Whenever it's clear, in the sequel we omit "non contradictory" and "non redundant" when we mention prime implicants.

2.1 Belief Revision Issue

Belief revision consists of inserting in a consistent way a new piece of information μ into a belief set ψ [6]. Revision operator is usually denoted by $\circ$ and the resulting belief set is denoted by $\psi \circ \mu$. The AGM postulates provide an axiomatic characterisation of belief revision operators [1,7]. In the context of finite propositional beliefs, AGM postulates can be rephrased as follows [12]:

(R1) $\psi \circ \mu$ implies μ.
(R2) If $\psi \wedge \mu$ is satisfiable then $\psi \circ \mu \equiv \psi \wedge \mu$.
(R3) If μ is satisfiable then $\psi \circ \mu$ is also satisfiable.
(R4) If $\psi_1 \equiv \psi_2$ and $\mu_1 \equiv \mu_2$ then $\psi_1 \circ \mu_1 \equiv \psi_2 \circ \mu_2$.
(R5) $(\psi \circ \mu) \wedge \phi$ implies $\psi \circ (\mu \wedge \phi)$.
(R6) If $(\psi \circ \mu) \wedge \phi$ is satisfiable then $\psi \circ (\mu \wedge \phi)$ implies $(\psi \circ \mu) \wedge \phi$.

As mentioned earlier, even if they have been widely accepted, these postulates are too weak to characterise minimal change with respect to relevant revision. For almost all belief revision operators, minimality is represented with the help of an extra logical criteria of distance between the initial belief set and the incoming information [10]. That is, revising ψ by μ consists of choosing the closest models of μ with respect to ψ [13,12]. Let $\preceq_\psi$ be a total pre-order representing preferences and defined over set $\mathcal{W}$ and representing the closeness criterion: $w \preceq_\psi w'$ states that w is at least as close as w' w.r.t. ψ. Faithful assignment represents preferences which are "centered" on ψ, i.e. the most preferred models are the models of ψ[1].

Theorem 1. *[13] Let $\mathcal{F}$ be a faithful assignment that maps each belief set ψ, to a total pre-order $\preceq_\psi$ over $\mathcal{W}$ such that the following three conditions hold:*

(C1) *if $w, w' \in [\![\psi]\!]$ then $w \not\prec_\psi w'$*
(C2) *if $w \in [\![\psi]\!]$ and $w' \notin [\![\psi]\!]$ then $w \prec_\psi w'$*
(C3) *if $\psi \equiv \varphi$ then $\preceq_\psi = \preceq_\varphi$*

A revision operator $\circ$ satisfies **(R1)–(R6)** *if and only if* $[\![\psi \circ \mu]\!] = \min([\![\mu]\!], \preceq_\psi)$

[1] $\prec_\psi$ is defined from $\preceq_\psi$ as usual, i.e., $w \prec_\psi w'$ iff $w \preceq_\psi w'$ but not $w' \preceq_\psi w$.

One of the simplest ways to set the preferences is to consider the propositional symbols that may change. This has been proposed by Dalal in [5]. It consists of characterising belief revision operator as a function which changes in each model of ψ the minimal number of propositional symbol truth values so that incoming information can be added without entailing inconsistency.

3 Prime Implicants Based Revision

Prime implicants enable us to define belief revision operators in a syntactic way while avoiding the issue of syntax dependency. This characteristic is due to the fact that each formula has only one set of prime implicants. The applicability of prime implicants and implicates in belief change area has been investigated and presented in several works, e.g [3,18,22,2,17], as well as the properties of prime implicants and prime implicates in [4]. We root our work in [3,17] which mainly focused on the notion of distance and the way to set preferences and extend these contributions to capture the notion of relevance.

3.1 Incorporating Prime Implicants

Given a belief set ψ and a new piece of information μ, let PI_ψ and PI_μ be the set of prime implicants of ψ and μ. In order to incorporate new information μ into the existing belief set ψ, we combine the prime implicants of ψ and μ in such a way that for every $D_\psi \in PI_\psi$ and $D_\mu \in PI_\mu$, a new term is obtained by adding to D_μ all the literals of D_ψ which are not conflicting with the literals to D_μ, as stressed by the following incorporating function Γ:

Definition 1. *Let* $\Gamma : \mathcal{L}(P) \times \mathcal{L}(P) \mapsto 2^{\mathcal{D}}$ *be a function defined as follows:*

$$\Gamma(\psi, \mu) = \{D_\mu \cup (D_\psi - \overline{D_\mu}) | D_\psi \in PI_\psi \text{ and } D_\mu \in PI_\mu\}$$

where PI_ψ *and* PI_μ *are the sets of prime implicants of* ψ *and* μ*.*

Intuitively, the set $\Gamma(\psi, \mu)$ contains all the terms that are obtained by extending each prime implicant of μ with the maximal consistent part of each prime implicant of ψ.

Example 1. Consider the following sets of prime implicants: $PI_\psi = (\neg p_2 \wedge \neg p_3) \vee (\neg p_2 \wedge p_4) \vee (\neg p_1 \wedge \neg p_3 \wedge p_4)$ and $PI_\mu = (p_3 \wedge \neg p_4) \vee (p_1 \wedge p_2)$. The following table presents the set of terms in $\Gamma(\psi, \mu)$:

We extend the previous definition with the set $\Gamma(\psi) \subseteq \mathcal{D}$ s.t. $\Gamma(\psi) = \bigcup_{D \in \mathcal{D}} \Gamma(\psi, D)$ which denotes the set of all terms that can be defined according to ψ (i.e. all possible consistent μ are considered).

3.2 Preference Ordering over Terms

To construct a belief revision operator using prime implicants, we need to set preferences over terms instead of worlds. Let $\leqslant_\psi$ be a preference relation defined over the set of possible terms $\mathcal{D}$: $D \leqslant_\psi D'$ states that D is at least as close as D' w.r.t. ψ. As for preferences set over worlds, we define the notion of faithful assignment.

Table 1. Incorporating prime implicants of two formulas

D_ψ	D_μ	$D_i \in \Gamma(\psi, \mu)$	
$\neg p_2 \wedge \neg p_3$	$p_3 \wedge \neg p_4$	$\neg p_2 \wedge p_3 \wedge \neg p_4$	(D_1)
$\neg p_2 \wedge \neg p_3$	$p_1 \wedge p_2$	$p_1 \wedge p_2 \wedge \neg p_3$	(D_2)
$\neg p_2 \wedge p_4$	$p_3 \wedge \neg p_4$	$\neg p_2 \wedge p_3 \wedge \neg p_4$	(D_1)
$\neg p_2 \wedge p_4$	$p_1 \wedge p_2$	$p_1 \wedge p_2 \wedge p_4$	(D_3)
$\neg p_1 \wedge \neg p_3 \wedge p_4$	$p_3 \wedge \neg p_4$	$\neg p_1 \wedge p_3 \wedge \neg p_4$	(D_4)
$\neg p_1 \wedge \neg p_3 \wedge p_4$	$p_1 \wedge p_2$	$p_1 \wedge p_2 \wedge \neg p_3 \wedge p_4$	(D_5)

Definition 2. *A* faithful assignment $\mathcal{F}$ *is a function which maps every formula* ψ *to a pre-order over* $\Gamma(\psi)$ *such that*[2]*:*

(C1-T) *if* $D_u, D_v \in \mathcal{D}_\psi$*, then* $D_u \not<_\psi D_v$.
(C2-T) *if* $D_u \in \mathcal{D}_\psi$ *and* $D_v \notin \mathcal{D}_\psi$*, then* $D_u <_\psi D_v$.
(C3-T) *if* $\psi \equiv \varphi$*, then* $\leqslant_\psi = \leqslant_\varphi$.
(CI-T) *For all* $D_u \notin D_\psi, D_v \notin D_\psi$*, if* $(D_u \subseteq D_v)$ *then* $D_u \sim_\psi D_v$.

The first key difference between the two notions of faithful assignment is the domain used for preferences: it is required that preferences have to be defined on a subset of terms rather than on the whole set of possible worlds. That is, the pre-order is only required to be set over the set of terms that can be built from ψ and function Γ. The three constraints **(C1-T)**–**(C3-T)** are similar to the constraints **(C1)**–**(C3)**. The second key difference is the constraint **(CI-T)** which states that preferences should not favour too specific terms.

Example 2. Suppose $\psi = p_1 \wedge \neg p_2$; suppose two terms $\neg p_1 \wedge \neg p_2$ and $\neg p_1 \wedge \neg p_2 \wedge p_3$ which belong to $\mathcal{D}_\psi$. Suppose that $\neg p_1 \wedge \neg p_2 \wedge p_3 <_\psi \neg p_1 \wedge \neg p_2$. It then means that if ψ is revised by $\neg p_1 \wedge \neg p_2$ then $\neg p_1 \wedge \neg p_2 \wedge p_3$ will be preferred to $\neg p_1 \wedge \neg p_2$ and thus the resulting belief set might contained extra and irrelevant information (p_3). Following the intuition of Parikh, relevance entails to focus changes on p_1 and p_2 and thus it cannot be the case that $\neg p_1 \wedge \neg p_2 \wedge p_3 <_\psi \neg p_1 \wedge \neg p_2$.

Hence, **(CI-T)** is a first step towards the enforcement of the notion of relevance.

Example 3. Let us pursue example 1. Suppose a faithful assignment such that $\leqslant_\psi$ is a preference ordering over the set of terms $\Gamma(\psi)$ based on Dalal's distance [5]. All terms $D \in \Gamma(\psi)$ are defined as follows: $D = D_\mu \cup (D_\psi - \overline{D_\mu})$ s.t. $D_\mu \in \mathcal{D}$ and $D_\psi \in PI_\psi$. $D \leqslant_\psi D'$ if and only if either (i) $D \in \mathcal{D}_\psi$ or (ii) the number of literals in the set $D_\psi \cap \overline{D_\mu}$ is less or equal to the number of literals that belongs to $D'_\psi \cap \overline{D'_\mu}$. As we can see, this pre-order is total over $\Gamma(\psi)$. It is straightforward to check that constraint **(C1-T)** and **(C2-T)** hold. Next, since the definition of prime implicants entails $\psi \equiv \varphi$ iff $PI_\psi = PI_\varphi$, constraint **(C3-T)** also holds. Finally, because Dalal's distance focuses on contradicting symbols, constraint **(CI-T)** also holds. Let us focus on the terms belonging to the set $\Gamma(\psi, \mu)$ shown in table 1. We get the following ordering over the terms in $\Gamma(\psi, \mu)$: $D_1 \sim_\psi D_2 \sim_\psi D_3 \sim_\psi D_5 <_\psi D_4$.

[2] $D \sim_\psi D'$ stands for $D \leqslant_\psi D'$ and $D' \leqslant_\psi D$.

3.3 Prime Implicant Based Revision

We are now able to define belief revision operators using prime implicants. A PI revision of ψ by μ is denoted by $\psi \circ_{PI} \mu$. Let us now characterise the PI revision operator:

Theorem 2. *Let $\mathcal{F}$ be a faithful assignment over $\mathcal{D}$ that maps each belief set ψ a total pre-order $\leqslant_\psi$. The PI revision operator $\circ_{PI}$ defined by $\mathcal{F}$ satisfies* **(R1)**–**(R6)** *if*

$$\psi \circ_{PI} \mu =_{def} \bigvee \min(\Gamma(\psi, \mu), \leqslant_\psi)$$

where $\min(\Gamma(\psi, \mu), \leqslant_\psi) = \{D \in \Gamma(\psi, \mu) \mid \forall D' \in \Gamma(\psi, \mu)$ *and* $D \leqslant_\psi D'\}$.

The proof is mainly based on [13], notice that constraint (**CI-T**) enable to enforce postulates (**R5**) and (**R6**)[3].

Example 4. According to the preferences detailed in the previous example, we get that terms D_1, D_2, D_3 and D_5 are minimal and compound the revised belief base:

$$\psi \circ_{PI} \mu = (\neg p_2 \wedge p_3 \wedge \neg p_4) \vee (p_1 \wedge p_2 \wedge \neg p_3) \vee (p_1 \wedge p_2 \wedge p_4) \vee (p_1 \wedge p_2 \wedge \neg p_3 \wedge p_4)$$

Notice that the PI revision is more restricted than the AGM revision since AGM revision considers all possible worlds while *PI* revision only focuses on terms belonging to $\Gamma(\psi)$.

Example 5. Let $\psi = p_1 \wedge p_2$ and $\mu = \neg p_1$. It is easy to see that $PI_\psi = \{p_1 \wedge p_2\}$ and $PI_\mu = \{\neg p_1\}$. Therefore $\psi \circ_{PI} \mu = \neg p_1 \wedge p_2$, no matter what the preference over the set of terms is. However, there are more than one AGM revision outcomes. Let $\mathcal{W} = \{11, 10, 01, 00\}$ be the set of interpretations of the language $\mathcal{L} = \{p_1, p_2\}$. If the faithful order of ψ over $\mathcal{W}$ is $11 \prec_\psi 10 \sim_\psi 01 \sim_\psi 00$, the outcome of the revision will be $[\![\psi \circ \mu]\!] = \{01, 00\} = [\![\neg p_1]\!]$. If the faithful order is $11 \prec_\psi 10 \sim_\psi 01 \prec_\psi 00$, then we have $[\![\psi \circ \mu]\!] = \{01\} = [\![\neg p_1 \wedge p_2]\!]$. Notice that p_2 cannot be changed by the revision of ψ with $\neg p_1$ in the prime implicant based revision; while p_2 may change with the AGM revision.

Let us now relate $\circ$ and $\circ_{PI}$ by showing that if the preferences over terms are linked to the preferences overs worlds, then theorems 1 and 2 are similar. That is the revised belief sets are equivalent whether we use worlds or terms. The following constraint (**KP**) states that preferences over terms and worlds have to be closely connected; i.e. if a term D_u is preferred to a term D_v then we have the same preferences between the worlds in which these two terms are satisfied. Let $\leqslant_\psi$ be a faithful pre-order over $\mathcal{D}$ and $\preceq_\psi$ be a faithful pre-order over $\mathcal{W}$ associated with ψ such that:

$$D_u \leqslant_\psi D_v \iff \forall u \in [\![D_u]\!], \exists v \in [\![D_v]\!] \; u \preceq_\psi v \qquad \textbf{(KP)}$$

Let us now relate operators $\circ$ and $\circ_{PI}$ by stating that if constraint (**KP**) is satisfied then both operators give similar results.

[3] Due to space restrictions all proofs have been omitted. A longer version of the paper which includes all the proofs is downloadable at the URL http://www.irit.fr/~Laurent.Perrussel/ai10-long.pdf.

Theorem 3. *Let $\preceq_\psi$ and $\leqslant_\psi$ be two faithful assignments over $\mathcal{W}$ and $\Gamma(\psi)$. Revision operators $\circ$ and $\circ_{PI}$ produce identical belief sets, that is $[\![\psi \circ_{PI} \mu]\!] = [\![\psi \circ \mu]\!]$ if and only if for all $D_u, D_v \in \Gamma(\psi)$ constraint (**KP**) is satisfied.*

The previous theorem confirms that operator $\circ_{PI}$ describes a specific family of AGM revision operators; that is, combined with additional constraints, postulates (**R1**)–(**R6**) also characterise PI revision. In the next section, we show that this specific aspect (or additional constraints) is in fact rooted in the notion of relevant revision.

4 Relevant Revision

The common shared opinion for setting the notion of relevant revision is to ground this notion into the languages used for describing belief bases [19,9,20,16]. If a statement φ in the belief base ψ does not share any propositional symbols with incoming information μ, then φ should belong to the resulting belief base. Parikh proposes the following postulate to capture the idea of relevant revision [19]:

(P) Let $\psi = \varphi \wedge \varphi'$ such that $\mathsf{Lang}(\varphi) \cap \mathsf{Lang}(\varphi') = \emptyset$. If $\mathsf{Lang}(\mu) = \mathsf{Lang}(\varphi)$, then $\psi \circ \mu \equiv (\varphi \circ' \mu) \wedge \varphi'$, where $\circ'$ is the revision operator restricted to language $\mathsf{Lang}(\varphi)$.

In general it is not easy to split the irrelevant statements from the belief base because the syntactical representation of the belief base could "falsify" us [16]. However, if we represent the belief set in prime implicants, this splitting becomes much more visible. This motivates us to rephrase Parikh's postulate in terms of the prime implicant representation of formulas:

(P-T) Let $\psi = \varphi \wedge \varphi'$. If $\mathsf{Lang}(PI_\varphi) \cap \mathsf{Lang}(PI_{\varphi'}) = \emptyset$ and $\mathsf{Lang}(PI_\mu) = \mathsf{Lang}(PI_\varphi)$, then $\psi \circ_{PI} \mu \equiv (\varphi \circ'_{PI} \mu) \wedge \varphi'$, where $\circ'_{PI}$ is the revision operator restricted to the language $\mathsf{Lang}(PI_\varphi)$.

However, the construction of $\circ_{PI}$ is not sufficient for enforcing postulate **(P-T)**. As stressed in [20], the local revision operator mentioned in postulate **(P)** has to be context-independent. Suppose that there are two belief sets ψ and ψ' such that $\psi \equiv \varphi \wedge \varphi'$, $\psi' \equiv \varphi \wedge \varphi''$, $\mathsf{Lang}(PI_\varphi) \cap \mathsf{Lang}(PI_{\varphi'}) = \emptyset$ and $\mathsf{Lang}(PI_\varphi) \cap \mathsf{Lang}(PI_{\varphi''}) = \emptyset$. Then there should exist only one single version of the local revision operator $\circ'$ such that $\psi \circ \mu \equiv (\varphi \circ' \mu) \wedge \varphi'$ and $\psi' \circ \mu \equiv (\varphi \circ' \mu) \wedge \varphi''$ for any μ s.t. $\mathsf{Lang}(PI_\mu) \subseteq \mathsf{Lang}(PI_\varphi)$. We also agree for this reading of postulate **(P)** qualified by [20] as the *strong* version of postulate **(P)**. Let us represent this notion in our framework. Assume that $\psi \equiv \varphi \wedge \varphi'$ s.t. $\mathsf{Lang}(PI_\varphi) \cap \mathsf{Lang}(PI_{\varphi'}) = \emptyset$. Having one local revision operator means that we have only one pre-order $\leqslant_\varphi$ associated to φ. Now, let us suppose two terms D and D' such that $D \leqslant_\varphi D'$. Pre-order $\leqslant_\psi$ should also reflect these preferences; that is extending terms D and D' with any prime implicants belonging to $PI_{\varphi'}$ will not change the preferences. The following constraint states this by saying how we can switch from one pre-order to a second one.

(PS-T) Let $\psi \equiv \varphi \wedge \varphi'$ such that $\mathsf{Lang}(PI_\varphi) \cap \mathsf{Lang}(PI_{\varphi'}) = \emptyset$. For any $D, D' \in \Gamma(\varphi)$: $D \leqslant_\varphi D'$ iff $D \cup D_{\varphi'} \leqslant_\psi D' \cup D'_{\varphi'}$ such that $D_{\varphi'}, D'_{\varphi'} \in PI_{\varphi'}$ and $D \cup D_{\varphi'}, D' \cup D'_{\varphi'} \in \Gamma(\psi)$.

In other words, this constraint expresses the strong notion of relevance by considering multiple faithful assignments. Now, we conclude that operator $\circ_{PI}$ characterizes relevant belief revision by satisfying postulate **(P)**. We first show that satisfying constraint **(PS-T)** entails that operator $\circ_{PI}$ satisfies the relevance postulate.

Theorem 4. *If faithful assignment $\leqslant_\psi$ satisfies **(PS-T)** then $\circ_{PI}$ postulate* **(P-T)**.

The theorem shows that the relevance is rooted in two key aspects: the definition of the revision operator $\circ_{PI}$ and the commitment to the strong version of relevance postulate.

Now, let us look at the opposite way. The question is: is operator $\circ_{PI}$ too restrictive or not? That is, if a revision operator $\circ$ satisfies postulate **(P)**, then can we exhibit an operator $\circ_{PI}$ which produces the same result? If the answer is positive then it means that in fact operator $\circ_{PI}$ characterizes the family of belief revision operators that satisfy postulate **(P)**. The following theorem shows that it is in fact the case under the condition that we focus on strong meaning of relevance.

Theorem 5. *Suppose a revision operator $\circ$ such that **(R1)–(R6)** and **(P)** hold. There always exists an operator $\circ_{PI}$ such that **(PS-T)** holds for $\circ_{PI}$ and $\psi \circ \mu \equiv \psi \circ_{PI} \mu$.*

The theorem tells us that $\circ_{PI}$ can represent operators for relevant belief change. Indeed, in [19,20] it has been shown that we can always define an operator $\circ$ which satisfies postulates **(R1)**–**(R6)** and **(P)**. In other words, theorems 2, 4 and 5 show that prime implicant based revision operator exactly characterizes the notion of relevant belief revision: **(R1)**–**(R6)** and **(P)** $\iff$ $\circ_{PI}$.

Example 6. It can be easily shown that condition **(PS-T)** holds for Dalal-based preferences introduced in the previous examples. The consequence of the previous theorems is that Dalal's revision operator is relevant. A second consequence is that preferences underlying a relevant revision can always be connected through constraint (**KP**).

5 Conclusion

In this paper we have proposed a general characterisation of relevant revision that states the family of operators for relevant belief change. We rephrased the relevance postulate in terms of prime implicants, which leads to the characterisation of relevant revision.The use of prime implicant representation not only provides a natural way to identify relevance in beliefs but also has advantages in computation [11]. Also our approach is syntax based. It avoids the syntax dependency, therefore possess the advantages of syntactical and semantical approaches.

Most of the work about the notion of relevant belief revision is based on language splitting. A splitting of a language with respect a belief set ψ is a partition of the language such that ψ can be resembled by a set of belief sets described in each sublanguage. Makinson in [14,15] shown that any AGM compliant contraction operator

which performs contraction on the belief set in each sub-language satisfies Parikh's postulate. That is, only partitions concerned by incoming information should be contracted. Makinson argues that there should exist a language-based dependence relation between the impacted partition and incoming information. He also argues this dependence relation should be rooted on the notion of canonical language. We actually follow a similar idea: the prime implicant set Γ focuses on the dependence relation and the $\circ_{PI}$ operation focuses on the splitting language. That is, prime implicants (and also prime implicates) ensure that we focus on the smallest language for describing ψ.

Peppas *et al.* proposed an other approach to relevant revision [20]. Their contribution is two fold. Firstly they proposed two conditions that are imposed on an AGM revision operator and guarantee the satisfaction of Parikh's postulate. Secondly, they proposes a set of semantic conditions based on the system of spheres model so that a belief revision operator that satisfies the semantic conditions is a relevant revision operator. As we have seen, our approach is purely syntax-based. Moreover, our framework embeds a significant part of the notion of relevance in the definition of the operator itself while existing approaches [15,20] consider the relevance with the help of additional constraints.

As future work, we want to pursue this characterization of relevance. That is, at this stage, relevance only describes what should not change. It does give a positive perspective by stating what should change. For instance, consider $PI_\psi = p_1 \wedge p_2 \wedge p_3$ and $PI_\mu = \neg p_1 \wedge p_2$. According to postulate **(P)**, we conclude that p_3 should remain unchanged. At the same time, only p_1 represents a disagreement point: at least p_1 should be changed. Prime implicants help us to to focus on the literals which represent disagreements, that is the set $\cup\{D_\psi \cap \overline{D_\mu} | D_\psi \in PI_\psi \text{ and } D_\mu \in PI_\mu\}$. This set represents in fact a lower bound for revision: "at least, what should change?" while postulate **(P)** describes the upper bound of relevant revision: "at most, what should change?". Our aim is to investigate the interactions between these bounds in order to characterize relevance from a positive perspective.

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: partial meet contraction and revision functions. J. of Symbolic Logic 50(2), 510–530 (1985)
2. Bienvenu, M., Herzig, A., Qi, G.: Prime implicate-based belief revision operators. In: Proc. of ECAI 2008, pp. 741–742 (2008)
3. Bittencourt, G., Perrussel, L., Marchi, J.: A syntactical approach to revision. In: Proc. of ECAI 2004, pp. 788–792. IOS Press, Amsterdam (2004)
4. Bittencourt, G.: Combining syntax and semantics through prime forms representation. Journal of Logic and Computation 18(1), 13–33 (2007)
5. Dalal, M.: Investigations into a theory of knowledge base revision: Preliminary report. In: Rosenbloom, P., Szolovits, P. (eds.) Pro. of AAAI 1988, vol. 2, pp. 475–479. AAAI Press, Menlo Park (1988)
6. Gärdenfors, P.: Knowledge in Flux: Modelling the Dynamics of Epistemic States. Bradford Books, MIT Press (1988)
7. Gärdenfors, P.: Belief revision: An introduction. In: Gärdenfors, P. (ed.) Belief revision, pp. 1–20. Cambridge University Press, Cambridge (1992)
8. Hansson, S.: A Textbook of Belief Dynamics. Theory Change and Database Updating. Kluwer, Dordrecht (1999)

9. Hansson, S., Wassermann, R.: Local change. Studia Logica 70(1), 49–76 (1998)
10. Herzig, A., Rifi, O.: Propositional belief base update and minimal change. Artificial Intelligence 115(1), 107–138 (1999)
11. Jackson, P.: Computing prime implicants. In: Stickel, M.E. (ed.) CADE 1990. LNCS (LNAI), vol. 449, pp. 543–557. Springer, Heidelberg (1990)
12. Katsuno, H., Mendelzon, A.: On the difference between updating a knowledge base and revising it. In: Allen, J.F., Fikes, R., Sandewall, E. (eds.) Proc. of KR 1991, pp. 387–394. Morgan Kaufmann, San Mateo (1991)
13. Katsuno, H., Mendelzon, A.: Propositional knowledge base revision and minimal change. Artificial Intelligence 52(3), 263–294 (1991)
14. Kourousias, G., Makinson, D.: Parallel interpolation, splitting, and relevance in belief change. J. Symb. Log. 72(3), 994–1002 (2007)
15. Makinson, D.: Propositional relevance through letter-sharing. Journal of Applied Logic 7, 377–387 (2009)
16. Makinson, D., Kourousias, G.: Respecting relevance in belief change. Análisis Filosófico 26(1), 53–61 (2006)
17. Marchi, J., Bittencourt, G., Perrussel, L.: Prime forms and minimal change in propositional belief bases. Annals of Math. and AI (2010)
18. Pagnucco, M.: Knowledge compilation for belief change. In: Proc. of the 19th Australian Joint Conf. on Artificial Intelligence, pp. 90–99. Springer, Heidelberg (2006)
19. Parikh, R.: Beliefs, belief revision, and splitting languages, vol. 2, pp. 266–278. Center for the Study of Language and Information, Stanford (1999)
20. Peppas, P., Chopra, S., Foo, N.: Distance semantics for relevance-sensitive belief revision. In: Dubois, D., Welty, C., Williams, M.A. (eds.) Proc. of KR 2004, pp. 319–328. AAAI Press, Menlo Park (2004)
21. Quine, W.V.O.: On cores and prime implicants of truth functions. American Mathematics Monthly 66, 755–760 (1959)
22. Zhuang, Z.Q., Pagnucco, M., Meyer, T.: Implementing iterated belief change via prime implicates. In: Orgun, M.A., Thornton, J. (eds.) AI 2007. LNCS, vol. 4830, pp. 507–518. Springer, Heidelberg (2007)

Belief Erasure Using Partial Imaging

Raghav Ramachandran, Abhaya C. Nayak, and Mehmet A. Orgun

Department of Computing, Macquarie University,
NSW-2109, Australia

Abstract. In this work we develop and examine a generalized account of David Lewis's imaging in the same way as how Jeffrey's Probability Kinematics generalizes Bayesian conditionalization. We show that this naturally leads to a rational account of probabilistic belief removal – a rather long standing problem in the area. It turns out that our generalization of imaging indeed is an account of probabilistic belief erasure as opposed to probabilistic belief contraction. In the process we also examine two other accounts of probabilistic belief removal which are rather novel and difficult to classify under the standard belief change taxonomy.

1 Introduction

Any artificial agent needs to maintain its beliefs and perform necessary changes in light of any received information. One well-known way of representing the "belief state" of an agent is to use a subjective probability function. A subjective probability function describes an individual's personal judgement as to how likely it is that a particular event will take place. Clearly, a mechanism is needed to update such a probability function in light of received information. Imaging, presented by David Lewis, is one such mechanism which changes the probability function upon learning that a certain proposition is true [14]. However there are cases in which a change is called for but there is no proposition in the language whose truth is guaranteed by the observation. Imaging is not capable of handling such observations. There is a need for a generalized account of imaging along the line that Jeffrey's account of probability kinematics [10] is a generalization of Bayesian Conditionalization. We motivate the need for such a generalization with the following example. This example is a variation of example given by Jeffrey to support his argument [10].

Example 1. *We concern ourselves with an agent working in a laboratory. The laboratory also employs an experimental robot which is capable of dyeing any piece of cloth violet or blue. The agent always keeps a green coloured cloth on its desk. On returning from a break, the agent finds the robot leaving the room. The agent rushes to observe the colour of the cloth on its desk, afraid that it might have been dyed. Due to poor lighting in the room the agent is unable to decide whether the colour of the cloth has been changed. The agent's observation makes it believe that the cloth could still be green. We further assume that at any particular instant the cloth has only one colour among green, blue or violet.*

J. Li (Ed.): AI 2010, LNAI 6464, pp. 52–61, 2010.

Let G represent the sentence that the cloth is green*, V represent* the cloth is violet *and B represent* the cloth is blue*. Now the initial probability function P is such that $P(G) = 1$, $P(V) = 0 = P(B)$. After the robot leaves the room the new probability function P' should be such that $P'(G) = 0.6$ and $P'(\neg G) = 0.4$.*

There are two aspects to the scenario portrayed in this example. (1) The probability function should be changed from P to P'. This corresponds to a Jeffrey-kind of generalization of imaging. (2) In the literature of belief change in a probabilistic framework, for instance [9,12], a sentence which has a subjective probability of 1 is considered a belief of the agent. In the above example G is initially a belief of the agent. The result of changing P to P' is such that G no longer has a probability of 1. Belief removal in a probabilistic setting corresponds to changing the probability associated with a sentence from 1 to less than 1. The above example, therefore, also motivates the study of belief removal in a probabilistic setting. The literature on belief change in a probabilistic setting has been confined to inclusion of new beliefs. The study of removal of beliefs in this setting has not received much attention. We realize that these two problems have the same solution. In this work we focus on presenting a generalization of imaging and also give an account of belief removal in a probabilistic framework.

We begin by outlining the deficiency of conditionalization and Jeffrey conditionalization that does not allow them to model the change in a probability function described in the example (section 2), arguing that a generalization of imaging is needed. We then present a variation of imaging that can be used to model the scenario presented in the example (section 3). This variation of imaging should be able to model the withdrawal of a belief in a probabilistic framework. We show that this is in fact the case (section 4). Furthermore, we present two more variations of imaging and briefly discuss their properties (section 5). We conclude with a brief discussion on some finer points related to the proposed variations of imaging (section 6).

2 Background

Let $\mathcal{L}$ be a finite propositional language of the agent and the set of all possible worlds (interpretations) over $\mathcal{L}$ be Ω. The subjective probability function of an agent is given by a probability distribution over the set of all possible worlds. The probability distribution defined over Ω is such that, $\sum_{\Omega} P(\omega) = 1$. We represent the set of worlds which have non-zero probability under the probability function P by $\|P\|$. For any world $\omega \in \Omega$ and a sentence $A \in \mathcal{L}$, let $\omega(A) = 1$ when $\omega \models A$ (i.e. ω is a model of A) and $\omega(A) = 0$ when $\omega \not\models A$. The probability of a sentence A is given by

$$P(A) = \sum_{\Omega}\{P(\omega) : \omega(A) = 1\}.$$

A sentence A is said to be consistent with the probability function P if and only if $P(A) > 0$ and similary a sentence A is said to be inconsistent with P if and only if $P(A) = 0$.

The processes such as conditionalization, Jeffrey conditionalization and imaging change the probability distribution over the set of all possible worlds according to the observation made by the agent. However, in case of conditionalization and Jeffrey conditionalization the worlds with zero probability have zero probability even after the change. When a particular world is initially considered not possible, i.e. has zero probability, no received information can change this view. Therefore the probability of a sentence with the initial probability of 1 or 0 cannot be changed by these processes. In the example scenario G has initial probability of 1 and hence the change portrayed in the example cannot be modelled using conditionalization or even Jeffrey conditionalization.

In the process of imaging the probability of a sentence with initial probability of 1 or 0 can be changed. Imaging was developed by Lewis to provide semantics for Stalnaker conditionals [14]. The agent can change its subjective probability function to include new evidence E using imaging. Imaging makes use of a system of comparative similarity relations over Ω with respect to every single world in Ω. A comparative similarity system is one in which for every possible world ω there exists a total pre-order relation $\leq_\omega$. In [13,15] Lewis presents a detailed discussion on the comparative similarity relation among the worlds. The relation $\mu \leq_\omega \upsilon$ means that the world μ is *at least as similar to the world* ω *as the world* υ. The minimal world in $\leq_\omega$ which is a model of E is the most similar E-world to ω. We denote the most similar world of ω which is also a model of E by $\omega^{\#}_{E}$.[1]

Suppose the agent learns the truth of a sentence E. The agent shifts the probability mass associated with each model of $\neg E$ to some model of E. The probability associated with $\omega \in [\neg E]$ is shifted to the minimal model of E with respect to the relation $\leq_\omega$.[2] Imaging P by E, denoted as $P^{\#}_{E}$, is given by

$$P^{\#}_{E}(A) = \sum_{\Omega} \{P(\omega) : \omega^{\#}_{E}(A) = 1\}. \tag{1}$$

Imaging has been proved useful in solving many well-known problems such as *the Monty Hall Problem* [5] and *the Sleeping Beauty Scenario* [4,7]. In [11,6] imaging is said to model belief update. Update is the process of inclusion of a new belief in a dynamic environment. In Example 1, the agent suspects that the robot has altered the state of the world. Therefore we need to use imaging to model the given example scenario. However in Example 1 the observation cannot be represented as learning the truth of any particular proposition. Therefore it is not possible to use imaging as it is. We propose a variation of imaging, namely partial imaging, which we show is capable of handling the example scenario.

[1] In [14] Lewis assumes that the similarity relation is such that there is only a single most similar E-world to every $\neg E$-world. There has been arguments against this assumption [9]. Withdrawing this assumption does not affect this work. But here we carry on with this assumption.

[2] $[E]$ denotes the set of models of the sentence E.

3 Partial Imaging

Consider a sentence E which is inconsistent with the probability function P, i.e. $P(E) = 0$. Suppose the observation made by the agent requires the probability of E to be enhanced. Just as in imaging, probability is moved around but neither created nor destroyed. When the probability of a sentence E is to be enhanced, there needs to be a shift of probability mass from some $\neg E$-worlds (where E is false) to some E-worlds (where E is true). In order to have both E and $\neg E$ consistent with the resulting probability function, i.e. both have positive probability, we propose that only a portion of the probability mass on every $\neg E$-world be shifted to respective most similar E-world. This share is assumed to be uniform for all the $\neg E$-worlds with positive prior probability. Let the agent's observation result in assigning probability a to the sentence E. We represent the result of the observation as a pair $\langle E, a\rangle$. Let P^p_E denote the new probability function after changing P to accommodate $\langle E, a\rangle$. We formulate this change to P as follows.

Suppose $P(\neg E) = 1$ and ω be a possible world in which $\neg E$ is true, i.e. $\omega \models \neg E$. Let ω lose a-share of the probability mass associated with itself to the closest E-world. Then $\omega^{\#}_E$ gains a-factor of the probability mass of ω. Hence $P^p_E(\omega^{\#}_E) = a \cdot P(\omega)$ and $P^p_E(\omega) = (1 - a) \cdot P(\omega)$. We propose that all the $\neg E$-worlds lose a-share of their initial probability to their closest E-worlds. When imaging P by sentence E all the probability associated with $\neg E$-worlds is shifted to their closest E-worlds. But in partial imaging only a-factor of the probability mass is shifted. The changed probability of $\neg E$ is

$$P^p_E(\neg E) = \textstyle\sum_\Omega\{(1 - a) \cdot P(\omega) : \omega(\neg E) = 1\} = (1 - a) \cdot P(\neg E) = 1 - a.$$

The changed probability of sentence E is

$$P^p_E(E) = \textstyle\sum_\Omega\{a \cdot P^{\#}_E(\omega) : \omega(E) = 1\} = a \cdot P^{\#}_E(E) = a.$$

The new probability of any sentence A is given by

$$P^p_E(A) = a \cdot P^{\#}_E(A) + (1 - a) \cdot P(A).$$

As this process involves a partial shift of the probability mass in comparison with imaging, we call this partial imaging. Partial imaging of P with respect to a sentence E on making an observation $\langle E, a\rangle$ results in P^p_E, which is given by

$$P^p_E = P^{\#}_E a P \tag{2}$$

where $P^{\#}_E$ is the image of P with respect to E.[3] It must be noted that when $a = 1$ we have $P^p_E = P^{\#}_E$.

In Example 1, the agent decides to reduce the probability of G from 1 to 0.6 and enhance the probability of $\neg G$ from 0 to 0.4 upon making its observation. We suggest that the required new probability function is P^p_E where $a = 0.4$ and

[3] $P = P_1 a P_2$ is read as $P = a \cdot P_1 + (1 - a) \cdot P_2$.

E is $\neg G$. The models of $\neg G$ are models of V or B. The probability mass lost by G-worlds is shifted to the respective most similar worlds, which are models of $B \vee V$. Thus, the sentence $B \vee V$ is assigned a probability of 0.4 under the resultant probability function and G has a probability of 0.6. For different comparative similarity relations different $\neg G$-worlds receive the transferred probability. Therefore partial imaging is capable of handling the scenario given in Example 1, while imaging and all forms of conditionalization fail.

Observation 1. *Partial imaging is homomorphic.*[4]

In [9], Gärdenfors showed that imaging is homomorphic.[5] Pearl, in [16], identifies *homomorphism* as the property which enables modelling actions using imaging. Observation 1 plays an important role in our discussion on the relation between partial imaging and belief erasure in Section 5. In the following section we model a belief removal function by partial imaging.

4 Probabilistic Removal Functions

A belief state of an agent broadly includes its beliefs, the logical relationships between the beliefs and information about the ordering or ranking of beliefs. Many belief change operations have been studied in the literature. They can be classified broadly into two types: inclusion of a new belief and removal of an existing belief. In a probabilistic framework the belief state of the agent is often represented by a subjective probability function. Most proposals for belief change in a probabilistic framework have concentrated on inclusion of new information to the belief state [2,12,3,8]. Removal of existing beliefs from the belief state in a probabilistic framework has not been given much attention. In [9] the contraction of belief A from the belief state P, represented by P_A^-, is defined as follows:[6]

$$P_A^- = PaP_{\neg A}^* \tag{3}$$

where $P_{\neg A}^*$ is the result of revision of the initial belief state by $\neg A$ and $0 < a < 1$. For different values of a (3) gives different contraction functions. Thus the defined probabilistic contraction looks very similar to (2). Equation (2) states that partial imaging is a combination of initial belief state P and the result of including a new belief E by imaging. In equation (3), contraction by A is given as a combination of the initial belief state and the result of revision by negation of A. As equations (2) and (3) are similar, we consider the possibility of partial imaging modelling a belief removal operation.

In [11,6] imaging is used to model belief update. Update corresponds to inclusion of a new belief in a dynamic environment. The belief removal operation

[4] The proofs for the results can be obtained by contacting the first author.

[5] A probabilistic change function $*$ is said to be homomorphic if and only if, for all probability functions P and sentence A, for every P_1, P_2 and $a \in [0,1]$ such that $P = P_1 a P_2$ it holds that $P_A^* = (P_1)_A^* a (P_2)_A^*$.

[6] Belief Contraction is a belief change operation defined formally in [1].

corresponding to update is called belief erasure [11]. When the agent considers the world to have changed or suspects the world to have changed, the agent might consider it not profitable to hold on to certain beliefs. The agent might consider removing these beliefs from its belief state. Such a belief change is termed belief erasure. Let A be a belief of the agent. We denote the erasure operator by $\ominus$ and $P_A^{\ominus}$ denotes the result of erasure of A from P. We translate the erasure postulates to a probabilistic setting as follows.

E1 If $P_A^{\ominus}(B) = 1$ then $P(B) = 1$.
E2 If $P(\neg A) = 1$ then $P_A^{\ominus} = P$.
E3 If $\nvdash A$, then $P_A^{\ominus}(A) < 1$.
E4 If $\vdash A \leftrightarrow B$ then $P_A^{\ominus} = P_B^{\ominus}$.
E5 $(P_A^{\ominus})_A^{+} = P$, where $+$ denotes expansion of the belief state.
E6 $(PaP')_A^{\ominus}$ is equivalent to $P_A^{\ominus} a (P')_A^{\ominus}$ for any a such that $0 \leq a \leq 1$.

E1 states that erasure does not add new beliefs to the belief state of the agent. Postulate **E2** presents the trivial case. Erasure by negation of a belief does not change the belief state. According to **E3**, erasure is successful as long as the sentence being erased is not a logical tautology. Postulate **E4** states that erasure preserves logical equivalence, i.e. when the content of two sentences being erased is the same, the result of erasure is also the same. **E5** is the recovery postulate. When the agent erases a belief from its belief state and then expands by the same sentence then it arrives at the initial belief state. It must be noted that the expansion is modelled by conditionalization. A mix of probability functions PaP', $0 \leq a \leq 1$, represents a disjunction of belief states represented by P and P' respectively. Postulate **E6** states that erasure of a belief from a disjunction of belief states results in the disjunction of erasure of belief from the disjuncts. Any function that satisfies postulates **E1** to **E6** is an erasure function. The following theorem shows that partial imaging is an erasure function.

Theorem 1. *A partial imaging function is a belief erasure function.*

We can define erasure in a probabilistic setting in terms of partial imaging as follows. Let P be the belief state of the agent and $\neg A$ be the belief being erased from the belief state. Then,

$$P_{\neg A}^{\ominus} = P_A^p. \tag{E*}$$

Here we have shown that partial imaging is capable of modelling belief erasure. In the following section we present more variations of imaging. These variations are also capable of modelling removal of a belief in a probabilistic framework.

5 Variants of Imaging

In imaging, the probability distribution is changed with respect to a sentence E by shifting all the probability mass associated with each world in Ω to the respective most similar E-world. In partial imaging, the probability distribution is changed by shifting only a share of the probability mass associated with each world. In this section we describe two more variations of imaging.

5.1 Selective Imaging

Selective imaging needs a function that chooses some worlds with positive probability in the given belief state P. When the belief state needs to be changed, instead of all worlds losing their initial probability only the selected worlds do. To choose the worlds which lose their probability upon change we use a selection function. We denote the selection function by $\mathcal{S}$. Let $\mathcal{P}$ be the set of all probability distributions over Ω.

Definition 1. *A function $\mathcal{S}$ from the set of all possible probability distributions is said to be a selection function if and only if $\mathcal{S} : \mathcal{P} \longrightarrow 2^{\Omega}$ and $\mathcal{S}(P) \subseteq \|P\|$.*

For a given probability distribution P, let s denote the conjunction of all the sentences that hold true in all the selected worlds, i.e. $s = \bigwedge\{B \in \mathcal{L} | \forall \omega \in \mathcal{S}(P), \omega \models B\}$.[7] Then $P(s)$ is the sum of the probability of the selected worlds, i.e. $P(s) = \sum_{\omega \in \mathcal{S}(P)} P(\omega)$.

Suppose A is a sentence such that $P(A) = 0$. Let the initial probability function P be changed by selective imaging with respect to a sentence A. We denote the probability distribution resulting from selective imaging with respect to A by P^s_A. The worlds in $\mathcal{S}(P)$ lose their probability to the corresponding most similar A-worlds.[8] When ω is a selected world, i.e. $\omega \in \mathcal{S}(P)$, then ω loses all its probability - $P^s_A(\omega) = 0$.[9] As there is a shift of probability mass among the possible worlds, each sentence in $\mathcal{L}$ loses some probability mass through some worlds and gains certain probability mass through other worlds. We denote the probability lost by any sentence with the help of a function $\mathcal{P}_{lost}$. For every sentence in the language $\mathcal{P}_{lost}$ returns a value in [0,1], $\mathcal{P}_{lost} : \mathcal{L} \longrightarrow [0, 1]$. $\mathcal{P}_{lost}(B)$ denotes the probability lost by B when some B-worlds are selected by $\mathcal{S}$. The probability lost by B is the sum of probability lost by the B-worlds which are selected. It is given by

$$\mathcal{P}_{lost}(B) = \sum_{\Omega} \{P(\omega) : \omega \in \mathcal{S}(P) \text{ and } \omega(B) = 1\} = P(s \wedge B). \tag{4}$$

The probability lost by $\neg A$ when the probability function P is changed by selective imaging with respect to A is the sum of the probability of the selected worlds, i.e. $\mathcal{P}_{lost}(\neg A) = P(s \wedge A) = P(s)$.

As the A-worlds closest to the selected worlds gain probability, some sentences gain probability. The probability gained by the sentences in the language is given by the function $\mathcal{P}_{gain}$. The function $\mathcal{P}_{gain}$ is such that $\mathcal{P}_{gain} : \mathcal{L} \longrightarrow [0, 1]$. The *probability gain* for a sentence B is denoted by $\mathcal{P}_{gain}(B)$ and is given by

$$\mathcal{P}_{gain}(B) = \sum_{\Omega} \{P(\omega) : \omega \in \mathcal{S}(P) \text{ and } \omega^{\#}_A(B) = 1\}. \tag{5}$$

[7] $[s] = \mathcal{S}(P)$.

[8] We use $\mathcal{S}$ instead of $\mathcal{S}(P)$ when the probability function being referred to is clear from the context.

[9] It should be noted that irrespective of the input received the same set of worlds are chosen to lose their probability. We discuss this contentious feature in section 6.

When changing P by selective imaging with respect to A, probability gained by sentence A is $\mathcal{P}_{gain}(A) = P(s) = \mathcal{P}_{lost}(\neg A)$. However $\neg A$ does not gain any probability, that is, $\mathcal{P}_{gain}(\neg A) = 0$. The probability of any sentence B in the changed probability distribution, P^s_A, is given by

$$P^s_A(.) = P(.) + \mathcal{P}_{gain}(.) - \mathcal{P}_{lost}(.). \tag{6}$$

The resultant probability of $\neg A$ is $P^s_A(\neg A) = P(\neg A) - P(s)$, as expected. As selective imaging is capable of reducing the probability associated with a belief, we look at the compatibility between an erasure function and selective imaging.

Theorem 2. *Selective imaging is a belief removal function that obeys the postulates* ***E2***, **E3** and **E4**.

Selective imaging by satisfying **E2** is successful in modelling removal of a belief. However it does not satisfy all postulates of erasure. We briefly discuss the implications of this theorem in section 6. We model the situation in Example 1 using selective imaging. The observation made dictates the withdrawal of belief in G and enhancing the probability of $\neg G$. The result of selective imaging by $\neg G$ is such that $P^s_{\neg G}(G) = P(G) - P(s)$. The observation made in the example is $< \neg G, 0.4 >$. Even though selective imaging is capable of removal of belief in G, it is not possible to satisfy the requirement $P^s_{\neg G}(\neg G) = 0.4$ with the current definition of the selection function. Therefore, selective imaging is not capable of modelling situations as in Example 1.

5.2 Selective Partial Imaging

Another variation of imaging called Selective partial imaging(SPI), is a combination of both selective and partial imaging. In this method a few worlds in $\|P\|$ are selected. These selected worlds lose a share of their probability. Suppose A is a sentence such that $P(A) = 0$. We denote the result of changing P by selective partial imaging(SPI) with respect to A by P^{sp}_A.

Just as in selective imaging, this method uses a selection function as given in Definition 1. The probability lost by the *selected worlds* is transferred to their corresponding closest A-worlds. Let a $(0 < a < 1)$ be the scale for probability lost by the selected worlds which is gained by their respective closest A-worlds. For each world $\omega \in \mathcal{S}(P)$, ω loses a-share of its initial probability, i.e. $a \cdot P(\omega)$ and retains the rest of its initial probability, i.e. $(1-a) \cdot P(\omega)$. Hence, the result of SPI with respect to A is given by,

$$P^{sp}_A(.) = P(.) + a \cdot \mathcal{P}_{gain}(.) - a \cdot \mathcal{P}_{lost}(.) \tag{7}$$

where $\mathcal{P}_{lost}(.)$ is as defined in equation (5) and $\mathcal{P}_{gain}(.)$ is as defined in equation (6). This gives $P^{sp}_A(\neg A) = P(\neg A) - a \cdot P(s)$.

Similar to the case of selective imaging, we look for compatibility between erasure postulates and SPI. Since this method is a combination of selective and partial imaging, it carries over the inability of selective imaging in satisfying the postulates of erasure. SPI is a weaker belief removal operation and this we note in the following theorem.

Theorem 3. *SPI is a belief removal function that obeys the postulates* **E1**, **E2**, **E3** and **E4**.

The implications of this theorem are discussed in the following section. We look to model the situation in Example 1 with SPI. It is possible to model the withdrawal of belief in G with SPI. However, similar to selective imaging SPI is not capable of changing the probability function such that the requirement $P^{sp}_{\neg G}(\neg G) = 0.4$ is met. Therefore, SPI is not exactly suited for modelling belief change in Example 1.

It is clear that these three variations of imaging are different belief removal functions. Having described three different variations of imaging, we compare the three belief removal operations that these variations correspond to. The following theorem presents the relation between these operations.

Theorem 4. *Let $\neg A$ be a belief in the belief state P. The belief set is denoted by $\mathcal{K}$.*[10] *Suppose the agent removes the belief $\neg A$ by partial or selective or selective partial imaging, the resultant belief sets represented by $\mathcal{K}^p_A$, $\mathcal{K}^s_A$ and $\mathcal{K}^{sp}_A$ respectively, are such that $\mathcal{K}^p_A \subseteq \mathcal{K}^{sp}_A \subseteq \mathcal{K}^s_A$.*

Partial imaging is the strongest belief removal function compared to the other two. When the agent withdraws belief in $\neg A$ by partial imaging method then the set of beliefs retained are also retained when the agent removes $\neg A$ by selective or SPI. We also find that beliefs retained in SPI are also retained when $\neg A$ is removed by selective imaging.

6 Discussion and Conclusion

In the previous section we described variations of probabilistic imaging. Here we discuss some finer points about these variations.

The selection function depends only on the initial belief state. A keen reader would observe that irrespective of the belief being withdrawn the selection function chooses the same set of worlds. The worlds which lose their initial probability are always the same. This can be explained as follows. A sentence is said to be a belief if it has a probability of 1. Hence, for any belief A we have $\|P\| \subseteq [A]$. Therefore, any world which is selected by the selection function is a model of the belief being removed. Since the choice for the selection function always rises from the set of models of the belief being withdrawn, different sets of worlds need not be chosen for different beliefs. Therefore our definition of selection function is independent of the belief being withdrawn. However, the worlds which gain the probability differ depending on the belief being withdrawn.

Selective imaging and SPI do not satisfy all the postulates of erasure. While SPI satisfies **E1**, selective imaging does not. This implies that belief withdrawal by selective imaging adds new beliefs to the belief state of the agent. However, only sentences that are consistent with the initial belief state can be added to the set of beliefs after a belief removal operation by selective imaging. Also both

[10] The belief set corresponding to the belief state P is defined as the following set, $\mathcal{K} = \{B \in \mathcal{L} | P(B) = 1\}$.

selective imaging and SPI together do not satisfy **E5** which means that belief removal by these two methods is not reversible by expansion. Both variations of imaging do not satisfy **E6** which is because they both are not homomorphic. We reserve further investigations on these operations for future work.

To conclude, we motivate the need to be able to accommodate an observation that need not be expressible as some true sentence in the language. Imaging is not capable of handling this. We have proposed a variation of imaging, namely partial imaging which can handle this scenario. We have shown that partial imaging is indeed an erasure function. We also present selective imaging and SPI which are capable of modelling removal of a belief but are not erasure functions.

Acknowledgements. We would like to acknowledge and thank Prof. David Makinson for his valuable comments and suggestions on this work.

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. Journal of Symbolic Logic 50, 510–530 (1985)
2. Boutilier, C.: On the revision of probabilistic belief states. Notre Dame J. Formal Logic 36(1), 158–183 (1995)
3. Chan, H., Darwiche, A.: On the revision of probabilistic beliefs using uncertain evidence. Artif. Intell. 163(1), 67–90 (2005)
4. Cozic, M.: Imaging and sleeping beauty: A case for double-halfers. International Journal of Approximate Reasoning (2009) (in press, corrected proof:–)
5. Cross, C.B.: A characterization of imaging in terms of popper functions. Philosophy of Science 67(2), 316–338 (2000)
6. Dubois, D., Prade, H.: Belief revision and updates in numerical formalisms: An overview, with new results for the possibilistic framework. In: IJCAI, pp. 620–625 (1993)
7. Elga, A.: Self-locating belief and the Sleeping Beauty problem. Analysis 60(2), 143–147 (2000)
8. Gärdenfors, P.: The dynamics of belief: Contractions and revisions of probability functions. Topoi 5(1), 29–37 (1986)
9. Gärdenfors, P.: Knowledge in Flux: Modeling the Dynamics of Epistemic States. Bradford Books, MIT Press, Cambridge Massachusetts (1988)
10. Jeffrey, R.C.: The Logic of Decision. The University of Chicago Press, Chicago (1983)
11. Katsuno, H., Mendelzon, A.O.: On the difference between updating a knowledge base and revising it. In: Gärdenfors, P. (ed.) Belief Revision, pp. 183–203. Cambridge University Press, Cambridge (1992)
12. Kern-Isberner, G., Rödder, W.: Belief revision and information fusion on optimum entropy. Int. J. Intell. Syst. 19(9), 837–857 (2004)
13. Lewis, D.: Counterfactuals. Harvard University Press, Cambridge (1973)
14. Lewis, D.: Probabilities of conditionals and conditional probabilities. The Philosophical Review 85(3), 297–315 (1976)
15. Lewis, D.: On the Plurality of Worlds. Basil Blackwell, Malden (1986)
16. Pearl, J.: From imaging and stochastic control to a calculus of actions. In: Symposium Notes of the 1994 AAAI Spring Symposium on Decision-Theoretic Planning, Stanford, CA, pp. 204–209 (1994)

The Impact of Qualification on the Application of Qualitative Spatial and Temporal Reasoning Calculi

Carl Schultz[1], Robert Amor[2], and Hans W. Guesgen[3]

[1] SFB/TR 8 Spatial Cognition, The University of Bremen, Germany
[2] The University of Auckland, New Zealand
[3] Massey University, New Zealand

Abstract. Ever since Allen introduced his qualitative interval algebra in 1983, the area of qualitative spatial and temporal reasoning (QSTR) has been motivated by potential application areas that require human-oriented, commonsense reasoning. Despite this, it is well recognised in the community that there are relatively few commercial applications that heavily employ QSTR calculi. In this paper we directly address this issue by establishing a theoretical foundation for describing, developing and analysing QSTR based applications. We present an analysis of QSTR calculus qualification and investigate the impact that qualification has on a QSTR application's reasoning properties such as completeness and soundness. Our definition of QSTR applications also provides software developers with a basic template to begin creating their own applications. Concrete examples of existing QSTR applications are used to demonstrate and motivate this research.

1 Introduction

Qualitative spatial and temporal reasoning (QSTR) calculi represent and reason about coarse, intuitive relations between objects. The most prominent QSTR calculus is Allen's Interval Algebra (IA) [1]; Allen defines thirteen jointly exhaustive, pairwise disjoint relations that can hold between a pair of temporal intervals: before, meets, overlaps, starts, during, finishes, after, met by, overlapped by, started by, contains, finished by, and equals. Allen's seminal contribution was to frame the problem of determining a minimally consistent qualitative description of object relations as a constraint satisfaction problem, and proposed a modified path-consistency algorithm that performs composition using a reference look-up table.[1] Allen's approach to qualitative temporal reasoning has motivated the development of a large number of QSTR calculi [2]. For example, Region Connection Calculus 8 (RCC8) defines eight topological relations that can hold between pairs of arbitrary regular regions [3]: disconnected (DC), externally

[1] Composition of two relations $R1$ and $R2$ produces the relation $R3$ such that for all x, y, z if $R1(x, y)$ and $R2(y, z)$ then $R3(x, z)$.

J. Li (Ed.): AI 2010, LNAI 6464, pp. 62–71, 2010.

connected (EC), partially overlaps (PO), tangential proper part (TPP), non-tangential proper part (NTPP), tangentially contains (TPPi), non-tangentially contains (NTPPi), and equals (EQ). As with IA, a reference table is used to implement the composition operator.

A qualitative representation of object relations is formulated as a constraint satisfaction problem in the following way [1]. A constraint network consists of a finite set of vertices (representing objects in the world) and directed edges between vertices (representing relations between objects). Each edge is a variable that contains a set of qualitative relations that can possibly hold between two objects. A network is non-atomic if at least one edge contains more than one possible relation, and a network is atomic if all edges contain exactly one relation. Allen's algorithm attempts to eliminate relations from each edge which are not consistent based on the composition of relations [1].

Most QSTR calculi come with (sometimes implicit) domains of interpretation [4], for example, a constraint network using IA relations is interpreted as a linear ordering W together with a subset U of the intervals $(w1, w2)$, $w1 < w2$ on W. The process of mapping a configuration in the domain of interpretation to a qualitative constraint network is *qualification*, which we denote as the relation map q. A network is consistent if, for each relation $R(x, y)$ there is some consistent instantiation of all objects in the domain of interpretation that also satisfies R. A network is path-consistent if, for all triples of variables x, y, z, any consistent instantiation of x, z can be extended to some consistent instantiation of y. A well recognised concept in QSTR is weak composition (or extensionality [5,6]). It was noticed that Allen's algorithm applied to different calculi does not always give path-consistency but algebraic closure, i.e. $\forall x, y, z \cdot R(x, y) \wedge S(y, z) \rightarrow T_1(x, z) \vee \ldots \vee T_n(x, z)$. In particular, the consequent is a necessary but not a sufficient condition as required by path-consistency. Therefore algebraic closure is weaker than path-consistency and some impossible relations may not be eliminated from the constraint network.

The development of QSTR calculi is very often motivated by potential application areas that require more coarse, intuitive reasoning. For example, Egenhofer developed qualitative approaches to facilitate querying in GIS [13], and Wolter et al. developed vessel navigation application using qualitative orientation [12]. However, it is well recognised in the community that there is an absence of commerical applications, developed by application domain experts, that heavily employ QSTR calculi [9,10]. Relatively little research in QSTR has focused directly on the issues faced by software developers that are interested in applying QSTR calculi. This paper addresses the needs of application developers by establishing a formal definition of QSTR applications and analysing the impact of qualification on the properties of QSTR application reasoning. Section 2 provides a formal definition of QSTR applications. Section 3 presents an approach to analysing QSTR calculi with respect to application requirements. Section 4 analyses completeness and soundness of QSTR applications based on the given definition. Section 5 presents the conclusions of the paper.

2 A General Definition of QSTR Applications

This section presents a general definition of QSTR applications that is simple and yet sufficient for modelling all existing QSTR applications that the authors are aware of, regardless of the application domain. The benefits of formally defining QSTR applications are firstly that the general properties of applications can be analysed in detail and secondly that it provides software developers with a basic template to begin creating their own applications. To demonstrate this a number of examples of existing QSTR applications are given using this formulation.

One approach that has been used to apply QSTR calculi is to treat network inconsistency as a metaphor for some undesirable condition which is specific to the application domain. For example, Nokel [11] (page 46, Figure 23) reasons about three valves attached to a tank where exactly one valve must be open at any time; an inconsistent network means invalid valve behaviour. There are three main problems with the metaphor approach. Firstly, by mixing application-specific criteria with general laws of spatial arrangements the meaning of consistency becomes overloaded and heavily dependent on the application. Secondly the metaphors can be awkward and difficult to implement without an advanced understanding of compositional reasoning. Thirdly the metaphor approach is restricted to constraining triples of binary relations which greatly reduces the expressiveness of possible QSTR applications.

We will now present our alternative definition of QSTR applications. We define QSTR applications as having either (a) numerical runtime input which is qualified, (b) qualitative runtime input, or (c) qualitative rules that determine how the application should respond to input. Using this basic formulation we characterise the general runtime behaviour as follows: (1) receive input (2) (optional) do qualification (3) construct constraint network (4) (optional) do consistency check (5) execute application rules (6) return output.[2] Formally a QSTR application is a function that maps input symbols to sets of output symbols (i.e. inferred expressions). The input is a set of expressions that represent premises (or facts) about the world description (or a sequence of n world descriptions); e.g., the input expressions can use numerical relations (e.g. *kitchen.x=550* and *kitchen.y=100*) or qualitative relations (e.g. *near(kitchen, livingRoom)*). The output is a set of expressions that is some relevant subset of all inferred facts (as defined by the developer); e.g., output symbol α may correspond to the expression $\exists x \in U \cdot bathroom(x) \wedge near(x, kitchen)$.

Definition 1. *Let T_i be a set of expressions in some input language, let O be a set of output symbols representing logical expressions, and let n be a natural number. A QSTR application A is a function $A(T_1, \ldots, T_n) \subseteq 2^O$.*

The rules used to infer new facts based on the input premises are first-order constraints. If a constraint is not satisfied then a new fact is inferred in order to

[2] In this paper we focus on purely qualitative applications. Hybrid applications integrate both qualitative and numerical rules.

satisfy the constraint. QSTR calculi typically represent uncertainty by maintaining disjunctions of relations that can *possibly* hold between two objects. Because QSTR applications infer new facts using QSTR calculi relations that *possibly* hold, then the application inferences produced must also be interpreted by the user as *possibly* holding. This formulation will now be used to define a number of existing QSTR applications.

SailAway [12]. Input is a configuration of vessels consistent with a qualitative orientation calculus T^{OPRA}. Rules T^X are a formalisation of maritime right-of-way rules that use simple custom qualitative relations such as *collisionAtRear* and vessel types such as *motorVessel*. Output is a (set of) sequences of $OPRA$ constraint networks that are consistent with T^X (i.e. simulates future vessel states and eliminates states that cause collisions).

GIS QueryBySketch[3] [13]. Input is a set of bitmap representations of objects parsed into vector representations. Qualification creates a constraint network with 9-Intersection relations, custom detailed topological relations, and cardinal directions (salient numerical information is also maintained). Output (of the qualitative module) is simply the constraint network with optional relaxation information such as neighbouring relations which is used by a query processor.

Tank-valves [11]. Input is a configuration consistent with T^{IA}. Output symbol γ indicates abnormal valve operation. Rules for generating γ are expressions where x,y,z are the intervals when valves are open: (1) $overlaps(x, y) \vee starts(x, y) \vee during(x, y) \vee finishes(x, y)$ (2) $before(x, z) \wedge \neg\exists y \cdot meets(y, z)$.

Lighting Design [14]. Input is a set of numerical and qualitative expressions about spatial objects consistent with a qualitative orientation calculus T^{BA} (block algebra). Rules formalise lighting principles (e.g. *brightAmbientIllumination*) and higher level principles about subjective impressions (e.g. *spaciousness*). Output is the subset of inferences that specify the subjective impressions of rooms.

3 Selecting QSTR Calculi

A critical role of the QSTR application developer is deciding which collection of QSTR calculi, if any, should be used. Important factors include complexity of reasoning and ontological requirements. However, even when a calculus is ontologically appropriate, the relations may be too coarse on their own to accurately model a particular application concept. We will now present an analysis of qualification to greatly assist in both the selection of QSTR calculi and the development of QSTR application rules. Qualification is critically important because when rules are defined using particular calculi the application is not able to distinguish between certain geometric configurations.[4] The power of this

[3] This is a hybrid application; we only emphasise the qualitative components.

[4] Qualification depends on the properties of QSTR calculi and is completely independent of the QSTR application formulation in the previous section.

analysis is that it integrates major areas of QSTR research [5,6,15,8] to give the developer direct insights into calculi under different key input conditions.

Determining whether a given constraint network is consistent is in general an intractable task [8]. Importantly, not all applications actually need to check the consistency of the input constraint networks as in many domains the input is guaranteed to be consistent (e.g. given by CAD tools). We will therefore firstly consider the case where input constraint networks are known to be consistent a-priori.

3.1 Qualification When Input Is a Consistent Atomic Network

We will firstly consider the case of atomic constraint networks that are known to be consistent a-priori.[5] Given two objects (for binary calculi) the pertinent information about distinguishing between geometric configurations is given directly by the qualification operator. For example, in IA *before(x,y)* is defined as $x^+ < y^-$ (where t^- and t^+ are the start and end points of interval t respectively), and in RCC8 *EC(x,y)* is defined as $x \cap y \neq \emptyset \wedge i(x) \cap i(y) = \emptyset$ (where $i(r)$ is the interior of region r). The developer simply needs to review the qualification operator (in practice, a software tool would be used to conduct this analysis). Determining qualification of three or more objects (for binary relations) is simply the conjunction of the qualifications of each individual pair of objects. Thus the set of geometric instances that satisfy $q(R_1(x_1, x_2))$ is a superset of $q(R_1(x_1, x_2) \wedge ... \wedge R_n(x_{2n-1}, x_{2n}))$. Importantly, if $R_1(x, y)$ in isolation is too coarse then by combining relations from different calculi the developer can design rules that more closely approximate the necessary distinctions [15].

If the developer knows two geometric configurations that need to be distinguished then they can easily determine whether some combination of relations is adequate.[6] A calculus can distinguish two geometric configurations that are described as a set T_i^c of numerical expressions if $q(T_1^c) \not\leftrightarrow q(T_2^c)$. Furthermore, the precise geometric configurations that are numerically distinct but qualitatively indistinguishable from a given geometric configuration T^c are given by the expression $\neg q(T^c) \wedge q^{-1}(q(T^c))$ (where q^{-1} gives the numerical expression that corresponds to the given qualitative expression, i.e. the inverse of qualification).

For example, a lighting designer wants to specify patches of light on walls where two spotlights are directed. Their specific criteria are that the entire wall is covered in light (with some light spillage) while no light patches overlap: (1) $Wall \subset L_1 \cup L_2$ (2) $\emptyset = i(L_1) \cap i(L_2)$. This is qualified in RCC as: (1) $EC(L_1, L_2)$ (2) $PO(Wall, L_1)$ (3) $PO(Wall, L_2)$. The unwanted geometric configurations are: $\neg q(T^c) \wedge q^{-1}(q(T^c)) = (Wall \not\subset L_1 \cup L_2) \wedge i(Wall) \cap i(L_1) \neq \emptyset \wedge i(Wall) \cap i(L_2) \neq \emptyset \wedge i(L_1) \cap i(L_2) = \emptyset$. The developer responds by adding a new relation $covered(w) \equiv \forall r \cdot C(w, r) \rightarrow \exists l \cdot light(l) \wedge C(r, l)$. However, *covered*

[5] The consistency check is not performed thus avoiding the difficulties that some QSTR calculi have in determining the consistency of atomic networks.

[6] Qualitative calculi are jointly exhaustive and so every operator used to describe a geometric configuration corresponds to one, or the disjunction of more than one, qualitative relation.

cannot be calculated because it refers to arbitrary regions r connected to the wall w that have not been stored in the constraint network. Thus, the developer includes another custom relation (computed during qualification before the constraint network is constructed): $covers(w, r) \equiv r = \bigcup\{r_i | C(w, r_i)\}$. Even though ontologically RCC is very useful this additional analysis makes it clear that RCC in isolation is not sufficient and helps to guide the developer in creating custom relations.

3.2 Qualification When Input Is a Consistent Non-atomic Network

We now consider the case of atomic constraint networks that are known to be consistent a-priori for which some ambiguous information is added making the network non-atomic (for example, the world is only partially observable). Due to weak composition some QSTR calculi cannot always determine when a relation is impossible to instantiate i.e. the constraint network is consistent but not path-consistent [6]. Thus the application may erroneously make inferences based on impossible relations; clearly this is critical information for the developer. The explicit non-extensional composition triads given by the QSTR community (e.g. [6]) can provide the developer with meaningful information about the practical limitations of the calculus and the behaviour of applications built using those calculi.

We will use the following example to demonstrate that, due to weak composition, a calculus cannot always detect impossible relations. An art director of The Gallery of the Accademia di Belle Arti in Florence wants to temporarily exhibit a notable sculpture. The permanent gallery centrepiece is Michelangelo's David and the director wants to create a natural flow from David onto the temporary exhibit to offer a unique, exciting perspective. However, the new exhibit should not be immediately adjacent in case it distracts from the initial impact of David. There is some ambiguity in the design as the director has not yet decided where to place the temporary exhibit. The developer encodes the following rules: (1) $adjacent(x, y) \equiv EC(x, y)$ (2) $accessible(x, y) \equiv \exists w \cdot walkpath(w) \wedge EC(w, x) \wedge EC(w, y)$ (3) $interference(x) \equiv \exists y \cdot exhibit(y) \wedge adjacent(x, y)$ (4) $surrounds(x, y) \equiv EC(x, y) \wedge convexHull(h, x) \wedge PP(y, h)$ (PP is *proper part*). The floor plans are qualified as follows: (1) $walkpath(W1)$ (2) $surrounds(W1, David)$. The new exhibit will be placed in the same space as David to get the flow on effect: $surrounds(W1, NewExhibit)$. The question is whether this RCC representation is adequate. Notice that in this context *surrounds* implies that one region fills the hole of another region; this is a well known case of non-extensional reasoning in RCC. Thus, the application alerts the director to the possibility that the subjective impression of David is interfered with by the new exhibit although this is impossible as David is *surrounded* by the walkpath $W1$. Without having a detailed understanding of RCC this can be puzzling and frustrating to a user. The problem is that: $EC(David, WP1) \wedge EC(NewExhibit, WP1)$, so RCC8 erroneously infers that $EC(David, NewExhibit)$ is possible giving $adjacent(David, NewExhibit)$ and $interference(David)$. Indeed, any non-extensional triads pose a possible trap

for the user and developer. After reviewing these triads (with the assistance of QSTR application development tools) the developer can determine that other relations in addition to RCC8 relations are required to formalise *surrounds*.

3.3 Qualification When Input Is an Inconsistent Network

In this section we analyse the case where the constraint network presented to the application is inconsistent in a way that is undetectable to the QSTR calculus. This covers a large number of real world application situations where there are conflicting sources of information about a world description. For example, a robot's sensory inputs may be in conflict, or eye-witness reports of the scene of a crime may disagree. Determining whether a constraint network is inconsistent is an intractable task, and thus there is the possibility that the application will process an inconsistent network without being notified by the QSTR calculi that the network is indeed inconsistent. It is highly important that the software developer knows how the application will behave if an inconsistent constraint network is undetected by the QSTR calculi, particularly for safety-critical applications.

In the previous two sections it was shown that the developer can determine exactly which world descriptions are indistinguishable, and thus they can predict the exact erroneous inferences that will be made. However, when networks are potentially inconsistent developers can no longer predict the exact erroneous inferences.[7]

Proposition 1. *Any QSTR application inference can be made when a network is undetectably inconsistent.*

Proof. Let N be an inconsistent network that is determined to be consistent by some calculus Q. Let e be an expression (constraint) that causes application A to produce output α. Let N' be some network that satisfies e and for which its vertex set is disjoint with the vertex set of N (no objects in common), then $N'' = N' \cup N$ is undetectably inconsistent by Q and A will erroneously produce α. □

This is a highly relevant property of QSTR calculi and developers need to know about the types of networks in which reasoning is potentially erroneous. The developer can then determine whether the risk of faulty reasoning under those specific circumstances is acceptable or unacceptable.

Firstly we consider atomic networks. For some QSTR calculi (but not all) algebraic closure is sufficient for determining consistency in atomic networks, including IA and RCC8. However, Renz and Ligozat [5] have proven that calculi that do not have a property called *closure under constraints* are not always capable of detecting an inconsistent atomic network. This is precisely the information that a developer needs to determine when an application may potentially produce incorrect inferences. For example, an art director has over constrained

[7] This means that the software developer cannot identify any vulnerable rule. It is important to stress that this is a property of QSTR calculi and not the application formulation in this paper.

the qualitative location of a light source that they are trying to place in their gallery. However, due to the limitations of their particular chosen QSTR calculi the fact that the atomic network is inconsistent has remained undetected. Based on the location of the light, a number of faulty inferences are produced including a design warning that the light source is visible as people enter the room and thus distracts from the impact of the other exhibits. This causes the director to needlessly spend considerable effort and time moving the exhibits around the room to fine-tune the design when in fact the design is physically unrealisable.

Determining whether a non-atomic network is consistent is an NP-hard problem for all QSTR calculi [8]. However, some of the most prominent advances in QSTR research have been the identification of maximal tractable subsets of a given calculus [8,7]. The non-atomic networks for which an application may produce erroneous inferences is precisely the intractable subsets of the calculus, and it is therefore the information about intractable subsets that is critical to a developer. This concludes the analysis of QSTR calculi qualification and the possibility of an application producing erroneous inferences due to the qualification and consistency-checking limitations of QSTR calculi.

4 Properties of QSTR Applications

We will now use the qualification and consistency-checking properties of QSTR calculi presented in the previous sections to determine application completeness and soundness. A QSTR application is *complete* if the set of application inferences is always an improper superset of the set of correct inferences.

Proposition 2. *QSTR applications are complete.*

Proof. QSTR calculi are sound and therefore no correctly *possible* relations are eliminated. Thus, the constraint network contains the set of actual relations. QSTR applications make inferences based on all constraint network relations. Thus applications make all inferences that use the set of actual relations. □

Completeness is a very useful property of QSTR applications. It provides the user with some certainty in the context of intractable reasoning problems; they can be guaranteed that any inferences that the application did not make surely do not hold. A QSTR application is *sound* if the set of application inferences is always an improper subset of the set of correct inferences.

Proposition 3. *QSTR applications are not sound.*

Proof. Follows from the qualification analysis in Section 3. □

QSTR applications are not sound as a result of the properties of the underlying QSTR calculi. Thus, not all QSTR application inferences necessarily actually hold.[8] We will now show that in fact completeness only holds for inferences that

[8] This mirrors the properties of Allen's algebraic closure reasoning algorithm which is sound but not complete, thus leaving a set of *possible* relations which is a superset of the real set of *possible* relations.

require the constraint network to be consistent. Some applications may need to produce particular outputs (i.e. inferences) only when the network is found to be *inconsistent*. For example, when a robot detects an inconsistent network then it knows that something is wrong with its sensors and may need to execute a calibration routine to try to correct the sensor problem.

Proposition 4. *QSTR applications are neither complete nor sound for inferences on inconsistent networks.*

Proof. First, non-soundness by example. Let an application produce α if the network is both inconsistent and satisfies $EC(x, y) \wedge EC(y, z) \wedge EC(x, z)$. Let N be a non-atomic network that has been correctly identified as inconsistent by the application, which also contains $EC(A, B)$ and $EC(B, C)$ where region B completely fills a hole in A. RCC8 composition will erroneously infer that $EC(A, C)$ is a possible relation (i.e. this is an additional inconsistency that was *not* detected by the application) and the application will erroneously produce α. Second, non-completeness. Let N be an inconsistent non-atomic RCC8 network such that algebraic closure cannot detect the inconsistency. Then no inferences that hold in N when N is inconsistent will be produced. □

Importantly, soundness of QSTR application inferences on an inconsistent network obeys exactly the same principles as soundness of consistent network inferences. That is, if a network is found to be inconsistent then we can guarantee that the network is actually inconsistent, and thus the cases where the application inferences will be incorrect are exactly the same cases presented in Section 3. Completeness of inferences on inconsistent networks depends on the QSTR calculus correctly detecting when a network is inconsistent as discussed in Section 3.3.

5 Conclusions

We have presented research that supports the development of applications that incorporate QSTR calculi. We define QSTR applications as functions that accept spatial and temporal information, execute domain specific rules, and then return relevant inferences as the application output. This definition is capable of expressing all existing QSTR applications that the authors are aware of, and four examples of existing QSTR applications were given. We analysed qualification by considering four distinct cases based on whether the constraint network is atomic or non-atomic, and whether the network is known to be consistent a-priori. This analysis integrates major areas of research from the QSTR community such as tractable subsets and weak-composition and identifies how developers can use this research to assist in selecting appropriate calculi and developing domain specific rules. The qualification results were then used to determine some basic properties of QSTR application reasoning. We showed that when QSTR applications make inferences using consistent networks they are complete but not sound, and when applications make inferences (intentionally) using inconsistent networks they are neither complete nor sound. An important area of future

research is the human-computer interaction issue of enabling developers to analyse qualification from application rules in a natural way (e.g. using graphical languages).

References

1. Allen, J.F.: Maintaining knowledge about temporal intervals. Communications of the ACM 26, 832–843 (1983)
2. Cohn, A.G., Renz, J.: Qualitative spatial reasoning. In: van Harmelen, F., Lifschitz, V., Porter, B. (eds.) Handbook of Knowledge Representation. Elsevier, Amsterdam (2007)
3. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. In: Proc. 3rd Int. Conf. on Knowledge Representation and Reasoning, pp. 165–176. Morgan Kaufmann, San Mateo (1992)
4. Ligozat, G., Mitra, D., Condotta, J.-F.: Spatial and temporal reasoning: beyond allens calculus. AI Communications 17(4), 223–233 (2004)
5. Renz, J., Ligozat, G.: Weak composition for qualitative spatial and temporal reasoning. In: van Beek, P. (ed.) CP 2005. LNCS, vol. 3709, pp. 534–548. Springer, Heidelberg (2005)
6. Li, S., Ying, M.: Region connection calculus: Its models and composition table. Artif. Intell. 145(1-2), 121–146 (2003)
7. Renz, J.: Maximal tractable fragments of the region connection calculus: A complete analysis. In: Dean, T. (ed.) IJCAI, pp. 448–455. Morgan Kaufmann, San Francisco (1999)
8. Nebel, B., Burckert, H.-J.: Reasoning about temporal relations: A maximal tractable subclass of allens interval algebra. J. ACM 42(1), 43–66 (1995)
9. Wallgrun, J.O., Frommberger, L., Wolter, D., Dylla, F., Freksa, C.: Qualitative Spatial Representation and Reasoning in the SparQ-Toolbox. In: Barkowsky, T., Knauff, M., Ligozat, G., Montello, D.R. (eds.) Spatial Cognition 2007. LNCS (LNAI), vol. 4387, pp. 39–58. Springer, Heidelberg (2007)
10. Hahmann, T., Gruninger, M.: Detecting physical defects: A practical 2d-study of cracks and holes. In: AAAI Spring Symposium on Benchmarking of Qualitative Spatial and Temporal Reasoning Systems, Palo Alto, Technical Report SS-09-02, pp. 11–16. AAAI Press, Menlo Park (2009)
11. Nokel, K.: Temporally Distributed Symptoms in Technical Diagnosis. LNCS, vol. 517. Springer, Heidelberg (1991)
12. Wolter, D., Dylla, F., Wolfl, S., Wallgrun, J.O., Frommberger, L., Nebel, B., Freksa, C.: Sailaway: Spatial cognition in sea navigation. In: 31st Annual German Conference on AI Advances in Artificial Intelligence, vol. 22(1), pp. 28–30. Springer, Kaiserslautern (September 2008)
13. Egenhofer, M.J.: Query Processing in Spatial-Query-by-Sketch. J. Vis. Lang. Comput. 8(4), 403–424 (1997)
14. Schultz, C.P.L., Amor, R., Lobb, B., Guesgen, H.W.: Qualitative design support for engineering and architecture. Advanced Engineering Informatics 23, 68–80 (2009)
15. Wolfl, S., Westphal, M.: On combinations of binary qualitative constraint calculi. In: Boutilier, C. (ed.) 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, pp. 967–973 (July 2009)

Two Methods for Constructing Horn Contractions

Zhi Qiang Zhuang and Maurice Pagnucco

National ICT Australia and
ARC Centre of Excellence in Autonomous Systems
School of Computer Science and Engineering
The University of New South Wales
{zqzhuang,morri}@cse.unsw.edu.au

Abstract. The *epistemic entrenchment* relation (EE-relation) over sentences and the *preference* relation (P-relation) over sets of sentences are used for constructing contraction operations in the AGM framework of belief change. The constructed contractions are the *epistemic entrenchment based contraction* (EEC) and the *transitively relational partial meet contraction* (TRPMC). Rott [1] establishes a close connection between the two relations and thus the corresponding contractions. It is shown that there is a one-to-one correspondence between TRPMC and EEC thus the two contractions perform identically in the sense that they can be characterised by an identical set of postulates. Recently, belief change for the Horn fragment of classical logic (i.e., Horn logic) has drawn much attention, resulting in several constructions for Horn contractions. In this paper we explore the connections between the two relations under Horn logic. We first define a Horn TRPMC by extending the partial meet Horn contraction defined in [2]. Our investigation shows that, unlike the result established by Rott [1], in Horn logic for each EE-relation there is a corresponding P-relation, but there exist P-relations that have no corresponding EE-relations. As a consequence the corresponding Horn EEC and Horn TRPMC perform differently.

1 Introduction and Background

Belief change literature has often grappled with the issue of computational efficiency. One recent focus of attention is the use of languages with limited expressiveness, in particular propositional Horn clauses, and several approaches to Horn contraction have resulted. Most of these approaches have adopted constructions based on maximal subsets of the reasoner's belief state that do not imply the sentence being removed (also known as *remainder sets*) [3,4,2,5]. An alternative construction based on the AGM epistemic entrenchment has been proposed [6]. *The aim of this paper is to explore the connection between remainder set constructions and epistemic entrenchment constructions by adapting a technique proposed by Rott [1].*

J. Li (Ed.): AI 2010, LNAI 6464, pp. 72–81, 2010.

The AGM framework [7] dominates the approaches in the area of belief change. It supposes an underlying language $\mathcal{L}$ associated with a consequence relation $\vdash$ and the generated consequence operation Cn is assumed to be Tarskian, compact, supraclassical and to satisfy the deduction theorem (for $X \subseteq \mathcal{L}$ and $\phi, \psi \in \mathcal{L}$, $\psi \in Cn(X \cup \{\phi\})$ iff $\phi \rightarrow \psi \in Cn(X)$). Clearly, the logic generated from $\mathcal{L}$ includes full propositional logic.

The operation of contraction in the AGM framework models the situation when beliefs are given up from the belief corpus of a rational agent. The belief corpus is modelled as a *belief set* which is a set of sentences closed under the consequence operation, that is if K is a belief set then $K = Cn(K)$.

If K is a set of sentences and ϕ a sentence, we write $K \downarrow \phi$ for the set of all maximal subsets X of K such that $\phi \notin Cn(X)$. If K is a belief set then elements of $K \downarrow \phi$ are called *remainder sets* of K with respect to ϕ. As noted in [7], $K \downarrow \phi$ is non-empty iff $\phi \notin Cn(\emptyset)$. A selection function γ associated with a set of sentences K is such that $\gamma(K \downarrow \phi)$ returns a non-empty subset of $K \downarrow \phi$ when $K \downarrow \phi$ is non-empty and returns K otherwise. Given a selection function γ for K, the *partial meet contraction* (PMC) $\dot{-}$ over K is defined as $K\dot{-}\phi = \bigcap \gamma(K \downarrow \phi)$. Morover $\dot{-}$ is a PMC iff it satisfies rationality postulates $(K\dot{-}1)$–$(K\dot{-}6)$ [7], which are regarded as the *basic postulates* for contraction.

$(K\dot{-}1)$ $K\dot{-}\varphi = Cn(K\dot{-}\varphi)$. (Closure)
$(K\dot{-}2)$ $K\dot{-}\varphi \subseteq K$. (Inclusion)
$(K\dot{-}3)$ If $\varphi \notin K$, then $K\dot{-}\varphi = K$. (Vacuity)
$(K\dot{-}4)$ If $\nvdash \varphi$, then $\varphi \notin K\dot{-}\varphi$. (Success)
$(K\dot{-}5)$ $K \subseteq (K\dot{-}\varphi) + \varphi$. (Recovery)
$(K\dot{-}6)$ If $\varphi \equiv \psi$, then $K\dot{-}\varphi = K\dot{-}\psi$. (Extensionality)
$(K\dot{-}7)$ $K\dot{-}\varphi \cap K\dot{-}\psi \subseteq K\dot{-}\varphi \wedge \psi$. (Conjunction overlap)
$(K\dot{-}8)$ If $\psi \notin K\dot{-}\varphi \wedge \psi$ then $K\dot{-}\varphi \wedge \psi \subseteq K\dot{-}\psi$. (Conjunction inclusion)

$(K\dot{-}7)$–$(K\dot{-}8)$ are regarded as *supplementary postulates* for contraction. A PMC satisfies the supplementary postulates if it is "transitively relational" in the sense that a transitive relation $\leqslant$ over all the remainder sets is used for guiding the selection function and thus the contraction. That is, only the most preferred remainder sets are selected for intersection. Given a belief set K and its associated relation $\leqslant$, the *transitively relational partial meet contraction* (TRPMC) $\dot{-}$ over K is defined as $K\dot{-}\phi = \bigcap\{M \in K \downarrow \phi \,|\, M' \leqslant M \text{ for all } M' \in K \downarrow \phi\}$ when $\phi \notin Cn(\emptyset)$ and $K\dot{-}\phi = K$ otherwise. It is shown in [7] that $\dot{-}$ is a TRPMC iff it satisfies $(K\dot{-}1)$–$(K\dot{-}8)$. In [1], the relation $\leqslant$ is extended to a preference relation over arbitrary sets of sentences—that is a relation over $2^{\mathcal{L}}$—and is referred to as a *P-relation*.

An *epistemic entrenchment* relation (EE-relation) reflects the relative entrenchment of sentences in $\mathcal{L}$ with respect to a belief set; the more entrenched, the more important a sentence. Intuitively, when forced to give up beliefs, a rational agent will choose the less entrenched one to give up. The EE-relation is used to guide us in performing an *epistemic entrenchment based contraction* (EEC).

Given a belief set K and its associated EE-relation $\leq$, an EEC $\dot{-}$ over K is given by $K\dot{-}\phi = K \cap \{\psi \,|\, \phi < \phi \vee \psi\}$ when $\phi \notin Cn(\emptyset)$ and $K\dot{-}\phi = K$ otherwise [8]. EE-relations have to satisfy certain constraints for the EEC to be characterised by $(K\dot{-}1)$–$(K\dot{-}8)$, as for TRPMC [8]. These conditions are as follows:

$(EE1)$ If $\varphi \leq \psi$ and $\psi \leq \chi$ then $\varphi \leq \chi$	(Transitivity)
$(EE2)$ If $\varphi \vdash \psi$ then $\varphi \leq \psi$	(Dominance)
$(EE3)$ $\varphi \leq \varphi \wedge \psi$ or $\psi \leq \varphi \wedge \psi$	(Conjunctiveness)
$(EE4)$ If $K \nvdash \bot$ then $\varphi \notin K$ iff $\varphi \leq \psi$ for every ψ	(Minimality)
$(EE5)$ If $\varphi \leq \psi$ for every φ then $\vdash \psi$	(Maximality)

Put simply, $(EE1)$–$(EE5)$ requires EE-relations to be transitive, to satisfy connectivity, to be such that tautologies are most entrenched and sentences not in K least entrenched.

2 Contractions under Horn Logic

A Horn clause is a clause that has at most one positive atom, eg. $\neg p \vee \neg q \vee r$. A Horn formula is a conjunction of Horn clauses. A Horn theory is a logically closed set of Horn formulas. Horn logic is generated from a fixed language $\mathcal{L}_H$ which consists of all the Horn formulas in $\mathcal{L}$, with standard model theoretic semantics. Unlike propositional logic as assumed by the AGM framework, Horn logic is not supraclassical and does not satisfy the deduction theorem. As a consequence, there is no contraction operation defined under Horn logic that satisfies the Recovery postulate $(K\dot{-}5)$ [9]. As in Delgrande and Wassermann [2] we identify an interpretation by the set of atoms true in that interpretation. For every $X \subseteq \mathcal{L}_H$, let $[X]$ be the set of models of X. For $\phi \in \mathcal{L}$ we write $[\phi]$ instead of $[\{\phi\}]$. A notable property of a Horn theory is that its models are closed under intersection of positive atoms [10,11]. Thus if H is a Horn theory, then for $M_1, M_2 \in [H]$ we have $M_1 \cap M_2 \in [H]$. Given a set of models X, we denote by $Cl_\cap(X)$ the closure of X under intersection of positive atoms. For any belief set K, by $M(K)$ we mean the set of all remainder sets of K, i.e., $M(K) = \bigcup\{K \downarrow \phi \,|\, \phi \notin Cn(\emptyset)\}$.

The notions of Horn subsets of a P-relation and an EE-relation are crucial in the sequel which is obtained from standard EE-relation and P-relation by subtracting the relations between non-Horn formulas and non-Horn sets.

Definition 1. *Let $\leqslant$ be a P-relation, its Horn subset $\leqslant_{\mathcal{H}}$ is defined as:*

$$M \leqslant_{\mathcal{H}} N \text{ iff } M \leqslant N \text{ and } M = Horn(M) \text{ and } N = Horn(N)\}$$

Definition 2. *Let $\leq$ be an EE-relation, its Horn subset $\leq_{\mathcal{H}}$ is defined as:*

$$\phi \leq_{\mathcal{H}} \psi \text{ iff } \phi \leq \psi \text{ and } \phi \in \mathcal{L}_H \text{ and } \psi \in \mathcal{L}_H$$

As advocated in [6], ideally a Horn contraction should be as plausible as AGM contractions. Since we are restricted to Horn logic, it is reasonable to conclude that a Horn contraction is as plausible as an AGM contraction if the two contractions perform identically in terms of Horn formulas. This idea is illustrated in Figure 1, where $\dot{-}_H$ is the Horn contraction and $\dot{-}$ is the standard AGM contraction. Given a belief set K, $Horn(K)$ is the set of all Horn formulas in K and furthermore if K is associated with an EE-relation $\leq$ or a P-relation $\leqslant$, then the corresponding EE-relation and P-relation associated with $Horn(K)$ is $\leq_{\mathcal{H}}$ and $\leqslant_{\mathcal{H}}$ respectively. To be more precise, we want to guarantee that for a belief set K and a Horn formula ϕ, the Horn contraction of ϕ from the Horn subset of K ($Horn(K)$) yields a resulting belief set ($Horn(K)\dot{-}_H\phi$)) that is exactly the Horn subset of the resulting belief set ($K\dot{-}\phi$) yielded by the AGM contraction of ϕ from K. Formally, such a Horn contraction is regarded as *Horn equivalent* to AGM contraction.

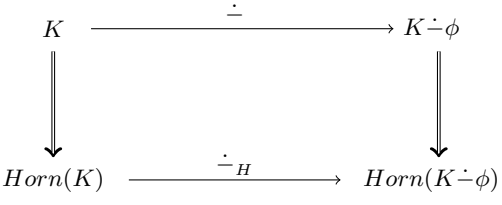


Fig. 1. Horn equivalence

Definition 3. *Let $\dot{-}$ and $\dot{-}_H$ be an AGM contraction and a Horn contraction respectively. $\dot{-}_H$ is Horn equivalent to $\dot{-}$ iff $Horn(K\dot{-}\phi) = Horn(K)\dot{-}_H\phi$, for all K and all $\phi \in \mathcal{L}_H$.*

As shown in [6], Horn contractions are not in general Horn equivalent to AGM contraction unless the associated EE-relation satisfies certain constraints, for instance the EE based Horn contraction in [6] imposes additional restrictions on the standard EE-relation (see below).

The *EE based Horn contraction* (EEHC) defined in [6] is an analogue of EEC. As noticed the original construction of EEC requires arbitrary disjunctions which may not be Horn formulas. To overcome this problem, [6] replaces non-Horn disjunction with its *Horn strengthenings* [12].

Definition 4. *[12] Given a clause φ, its set of Horn strengthenings, denoted by $\varphi_{\mathcal{H}}$ is such that $\varphi^H \in \varphi_{\mathcal{H}}$ iff φ^H is a Horn clause and there is no Horn clause φ' such that $\varphi^H \subset \varphi' \subseteq \varphi$.*

In Definition 4 clauses are considered as sets of atoms thus they can be compared set-theoretically. Essentially, Horn strengthenings of a non-Horn clause are obtained by removing all but one of its positive atoms. With the notion of Horn strengthening the EEHC is defined as follows:

Definition 5. *[6] If $\leq$ is an EE-relation over $\mathcal{L}_H$ (with respect to H) then the EEHC $\dot{-}$ also denoted as $C(\leq)$[1] is defined using the following condition.*

$$H\dot{-}\phi = \begin{cases} H \cap \{\psi \mid \exists \chi \in (\phi \vee \psi)_{\mathcal{H}} \text{ such that } \phi < \chi\} & \text{if } \nvdash \phi, \\ H, & \text{if } \vdash \phi. \end{cases}$$

[6] also provided a representation theorem for the EEHC.

Theorem 1. *[6] $\dot{-}$ is an EEHC iff it satisfies $(H\dot{-}1)$–$(H\dot{-}10)$.*

$(H\dot{-}1)$ $H\dot{-}\varphi = Cn(H\dot{-}\varphi)$.
$(H\dot{-}2)$ $H\dot{-}\varphi \subseteq H$.
$(H\dot{-}3)$ If $\varphi \notin H$ or $\vdash \varphi$, then $H\dot{-}\varphi = H$.
$(H\dot{-}4)$ If $\nvdash \varphi$, then $\varphi \notin H\dot{-}\varphi$.
$(H\dot{-}5)$ If $\psi \in H\dot{-}\varphi \wedge \psi$ then $\psi \in H\dot{-}\varphi \wedge \psi \wedge \delta$
$(H\dot{-}6)$ $\vdash \varphi \equiv \psi$, then $H\dot{-}\varphi = H\dot{-}\psi$.
$(H\dot{-}7)$ $H\dot{-}\varphi \cap H\dot{-}\psi \subseteq H\dot{-}\varphi \wedge \psi$.
$(H\dot{-}8)$ If $\psi \notin H\dot{-}\varphi \wedge \psi$ then $H\dot{-}\varphi \wedge \psi \subseteq H\dot{-}\psi$.
$(H\dot{-}9)$ If $\psi \in H$ and $\psi \notin H\dot{-}\phi$ then $\forall \chi \in (\phi \vee \psi)_{\mathcal{H}}$, $\chi \notin H\dot{-}\phi$
$(H\dot{-}10)$ If $\forall \chi \in (\phi \vee \psi)_{\mathcal{H}}$, $\chi \notin H\dot{-}\phi \wedge \psi$ then $\psi \notin H\dot{-}\phi$

$(H\dot{-}1)$–$(H\dot{-}8)$ are analogues of well known contraction postulates whereas $(H\dot{-}9)$ is derivable from $(K\dot{-}1)$–$(K\dot{-}8)$ but requires Recovery. $(H\dot{-}10)$ is specific for Horn contractions. EEHC is not Horn equivalent to AGM contraction unless the EE-relation satisfies $(EE6)$.

$(EE6)$ For each ϕ, there is a $\psi \in \phi_{\mathcal{H}}$ such that $\phi \leq \psi$.

$(EE6)$ requires any non-Horn formulas to be equally entrenched to its most entrenched Horn strengthenings. [6] shows that if the EE-relation $\leq$ an EEC is based on satisfies $(EE6)$, then the EEHC based on the Horn subset of $\leq$ (i.e., $\leq_{\mathcal{H}}$) is Horn equivalent to the EEC.

Theorem 2. *[6] Let $\leq$ be an EE-relation and $\leq_{\mathcal{H}}$ its Horn subset. If $\leq$ satisfies $(EE6)$ then the EEC $C(\leq)$ is Horn equivalent to the EEHC $C(\leq_{\mathcal{H}})$.*

3 Transitively Relational Partial Meet Horn Contraction

In this section we present a construction of Horn contraction that is analogous to the TRPMC. TRPMC relies on the notion of remainder sets and there have been several approaches to construct *partial meet Horn contractions* (PMHC) [3,4,2]. TRPMC is an extension of PMC such that TRPMC is guided by a P-relation over remainder sets. Similarly, the *Transitively Relational Partial Meet Horn Contraction* (TRPMHC) to be defined is an extension of the *partial meet Horn contraction* (for belief sets) in [2]. [2] proposed a notion of *weak remainder set*. Under propositional logic the remainder set of K with respect to ϕ can

[1] We sometimes use $C(\leq)$ instead of $\dot{-}$ to emphasise that this is an EEC or EEHC that is based on the EE-relation $\leq$.

be obtained semantically by adding a counter-model of ϕ to the models of K (i.e., $X \in K \downarrow \phi$ iff $[X] = [K] \cup \{m\}$ for $m \in [\neg\phi]$). Essentially, the number of remainder sets is identical to the number of counter-models of ϕ. Weak remainder sets are defined similarly, but taking into account the closure property of Horn theories (i.e., $[H] = Cl_{\cap}([H])$).

Definition 6. *[2] let H be a Horn belief set, and ϕ be a Horn formula. $H \downarrow_w \phi$ is the set of weak remainder sets such that $H' \in H \downarrow_w \phi$ iff there is a model $m \notin [\phi]$ such that $[H'] = Cl_{\cap}([H] \cup \{m\})$.*

Full negation is not available in Horn logic, for instance if $\phi = \neg p \wedge \neg q$ then $\neg\phi$ is not a Horn formula. The counter models of ϕ (i.e., $[\neg\phi]$) is defined to be any interpretation that is not a model of ϕ. Using the notion of weak remainder sets TRPMHC is defined as follows:

Definition 7. *If $\leqslant$ is a P-relation over $\mathcal{L}_H$ (with respect to H) then the TRPMHC $\dot{-}$ also denoted by $C(\leqslant)$[2] is defined using the following condition:*

$$H\dot{-}\phi = \begin{cases} \bigcap\{M \in H \downarrow_w \phi \mid M' \leqslant M \text{ for all } M' \in H \downarrow_w \phi\} & \text{if } \not\vdash \phi, \\ H, & \text{if } \vdash \phi. \end{cases}$$

It turns out that TRPMHC is more comprehensive than EEHC thus it admits more possible ways of removing Horn formulas. Theorem 3 which follows naturally from results in Section 4 justifies such connection between TRPMHC and EEHC.

Theorem 3. *If $\dot{-}$ satisfies $(H\dot{-}1)$–$(H\dot{-}10)$ then $\dot{-}$ is a TRPMHC.*

As in EEHC, we present the conditions for Horn equivalence of TRPMHC to AGM (TRPMC) contraction. Two technical lemmas are shown first.

Lemma 1. *If K is a belief set then $[Horn(K)] = Cl_{\cap}[K]$.*

Proof. $Horn(K) \subset K$ implies $[K] \subseteq [Horn(K)]$. Thus $Cl_{\cap}[K] \subseteq Cl_{\cap}[Horn(K)]$ which together with $[Horn(K)] = Cl_{\cap}[Horn(K)]$ implies $Cl_{\cap}[K] \subseteq [Horn(K)]$. Let H be a Horn belief set s.t. $[H] = Cl_{\cap}[K]$, so that $H \subset K$. Suppose $[Horn(K)] \not\subseteq Cl_{\cap}[K]$. So we have $[H] = Cl_{\cap}[K] \subset [Horn(K)]$ which implies $Horn(K) \subset H$. Thus there is $\phi \in \mathcal{L}_H$ s.t. $\phi \in H \subset K$ and $\phi \notin Horn(K)$, contradicting the definition of $Horn(\cdot)$. Hence $[Horn(K)] \subseteq Cl_{\cap}[K]$. □

Lemma 2. *Let K be a belief set and $\phi \in \mathcal{L}_H$. We have for all $M \in K \downarrow \phi$ that there exists $M_h \in Horn(K) \downarrow_w \phi$ such that $M_h = Horn(M)$.*

Proof. By a property of remainder sets, for each $M \in K \downarrow \phi$ we have $[M] = [K] \cup \{m\}$ for $m \in [\neg\phi]$. Let $[M'] = Cl_{\cap}([Horn(K)] \cup \{m\})$. By definition of weak remainder sets we have $M_h \in Horn(K) \downarrow_w \phi$. By Lemma 1, $[M_h] = Cl_{\cap}([Horn(K)] \cup \{m\}) = Cl_{\cap}(Cl_{\cap}[K] \cup \{m\}) = Cl_{\cap}([K] \cup \{m\}) = Cl_{\cap}[M]$. Again through Lemma 1 we have $M_h = Horn(M)$. □

[2] We sometimes use $C(\leqslant)$ instead of $\dot{-}$ to emphasise that this is a TRPMC or TRPMHC that is based on the P-relation $\leqslant$.

It follows from Lemma 2 that there is a one-to-one correspondence between standard remainder sets and weak remainder sets. Theorem 4 shows that if the Horn P-relation a TRPMHC is based on also has a one-to-one correspondence with the standard P-relation a TRPMC is based on, we will have a Horn equivalence between the TRPMHC and the TRPMC.

Theorem 4. *Let $\leqslant$ be a P-relation and $\leqslant_{\mathcal{H}}$ be its Horn subset. If $\leqslant$ satisfies $M \leqslant N$ iff $Horn(M) \leqslant Horn(N)$ for all logically closed M, N, then the TRPMHC $C(\leqslant_{\mathcal{H}})$ is Horn equivalent to the TRPMC $C(\leqslant)$.*

Proof. We write $\dot{-}_{\mathcal{H}}$ for $C(\leqslant_{\mathcal{H}})$ and $\dot{-}$ for $C(\leqslant)$. It suffices to show for any K and $\phi \in \mathcal{L}_H$, $Horn(K\dot{-}\phi) = Horn(K)\dot{-}_{\mathcal{H}}\phi$. If $\vdash \phi$ or $\phi \notin K$ then $Horn(K\dot{-}\phi) = Horn(K) = Horn(K)\dot{-}_{\mathcal{H}}\phi$ and we are done. So suppose $\nvdash \phi$ and $\phi \in K$. Let $X = \{M \in K \downarrow \phi | M^{'} \leqslant M$ for all $M^{'} \in K \downarrow \phi\}$ and let $X_h = \{M_h \in Horn(K) \downarrow_w \phi | M_h^{'} \leqslant M_h$ for all $M_h^{'} \in Horn(K) \downarrow_w \phi\}$. It follows from the properties of $\leqslant$ and Lemma 2 that for each $M \in X$ there is a $M_h \in X_h$ s.t. $M_h = Horn(M)$. Therefore we have $Horn(K\dot{-}\phi) = Horn(\bigcap\{M \in K \downarrow \phi | M^{'} \leqslant M$ for all $M^{'} \in K \downarrow \phi\}) = Horn(\bigcap X) = \bigcap X_h = \bigcap\{M_h \in Horn(K) \downarrow_w \phi | M_h^{'} \leqslant M_h$ for all $M_h^{'} \in Horn(K) \downarrow_w \phi\} = Horn(K)\dot{-}_{\mathcal{H}}\phi$. □

4 Connections between EEHC and TRPMHC

In the AGM framework TRPMC and EEC are characterised by an identical set of postulates [7,8]. Rott [1] investigated the connections between EEHC and TRPMHC by mapping the P-relation and EE-relation that determine the two contractions. Methods for obtaining one relation from the other are proposed and justified. The investigation shows that if a P-relation $\leqslant$ over a belief set K is obtained from an EE-relation $\leq$ over the belief set K then the EEC $C(\leq)$ returns identical belief sets as the TRPMC $C(\leqslant)$ for the contraction of any formula from K. Similar results also hold for P-relations and the obtained EE-relations. The results show essentially that there is a one-to-one correspondence between EEC and TRPMC and from which it can be concluded that EEC and TRPMC, although constructed differently, perform identically.

In this section we will explore connections between EEHC and TRPMHC by using the methodology in [1]. Our investigation reveals that for each EEHC there is a corresponding TRPMHC but not vice versa. It will become clear that the reason is that the Horn EE-relation, as a logical structure for capturing preference information, is not as expressive as the Horn P-relation.

Given an EE-relation, its corresponding P-relation is obtained by using Definition 8.

Definition 8. *[1] If $\leq$ is an EE-relation on $\mathcal{L}$ (with respect to a belief set K) then the P-relation $\leqslant$ on $2^{\mathcal{L}}$ (with respect to K), also denoted as $P(\leq)$, can be defined using the following condition:*

$$M \leqslant N \text{ iff for all } \phi \notin N \text{ there is a } \psi \notin M \text{ such that } \phi \leq \psi.$$

However, there exist Horn P-relations that are not obtainable from any Horn EE-relation via Definition 8. The following example illustrates one such P-relation.

Example 1. Let Horn belief set $H = Cn_H(\{\neg p \vee q, \neg q \vee r\})$. So we have $H \downarrow (\neg p \vee r) = \{①,②\}$, $H \downarrow (\neg p \vee q) = \{②,③\}$, $H \downarrow (\neg q \vee r) = \{①,④\}$, and $H \downarrow (\neg p \vee \neg q \vee r) = \{①\}$ for ①=$\{\neg p \vee q, \neg p \vee \neg r \vee q\}$, ②=$\{\neg q \vee r, \neg p \vee \neg r \vee q, \neg p \vee \neg q \vee r\}$, ③=$\{\neg q \vee r, \neg p \vee \neg q \vee r, \neg p \vee r\}$, and ④=$\{\neg p \vee q, \neg p \vee \neg r \vee q, \neg p \vee r, \neg p \vee \neg q \vee r\}$. Let P-relation $\leqslant$ be associated with H and contain ②=③=④>①. Suppose for contradiction that there is an EE-relation $\leq$ such that $\leqslant = P(\leq)$. It follows from ④>① and Definition 8 that $\neg p \vee \neg r \vee q > \neg q \vee r$. But it follows from ②=④ and Definition 8 that $\neg p \vee \neg r \vee q \leq \neg q \vee r$ so we have a contradiction and we can conclude $\leqslant$ can not be obtained from any EE-relation via Definition 8.

Given a P-relation, its corresponding EE-relation is obtained by using Definition 9.

Definition 9. *[1] If $\leqslant$ is a P-relation on $2^{\mathcal{L}}$ (with respect to a belief set K) then the associated EE-relation $\leq$ on $\mathcal{L}$ (with respect to K), also denoted as $E(\leqslant)$, is defined by the following condition:*

$$\phi \leq \psi \text{ iff for all } M \in M(K) \text{ such that } \psi \notin M, \text{there is an } M' \in M(K) \text{ such that } \phi \notin M' \text{and } M \leqslant M'.$$

Contrary to Horn P-relations, for any Horn EE-relation $\leq$ there is a Horn P-relation $\leqslant$ from which $\leq$ is obtainable via Definition 9. This follows directly from Theorem 1 of [1] which also holds under Horn logic. The theorem states if $\leq$ is an EE-relation then $\leq = E(P(\leq))$. Since P-relation $P(\leq)$ (obtained from $\leq$ via Definition 8) always exists, $\leq$ is always obtainable from $P(\leq)$ via Definition 9.

Example 2. Continuing with Example 1. Let $\leqslant_1$ contains ②=③=④>① and $\leqslant_2$ contains ②=③=④=①. Under propositional logic the EE-relation $E(\leqslant_1)$ and $E(\leqslant_2)$ only differ in the entrenchment of non-Horn clause $\neg p \vee q \vee r$. Since a Horn EE-relation contains only Horn formulas, under Horn logic we have $E(\leqslant_1) = E(\leqslant_2)$. The entrenchment information of $\neg p \vee q \vee r$ is not representable with Horn EE-relations but is still representable with Horn P-relations. Clearly, there is no EE-relation that captures the preference information in the P-relation $\leqslant_1$.

Based on the above analysis on the connections between Horn EE-relations and Horn P-relations, we claim that, under Horn logic EE-relations are not as effective as P-relations at representing preference information for performing contraction. P-relations provide more possible belief states than EE-relations.

In the remainder of this section, we will demonstrate how the connection between EE-relations and P-relations affect the EEHC and TRPMHC. As shown in [1], the EEC based on $\leq$ is equivalent to the TRPMC based on $P(\leq)$ (Theorem 4, [1]). It turns out that the equivalence also holds between EEHC and TRPMHC. We first give an important lemma.

Lemma 3. *Given an EE-relation $\leq$ and its Horn subset $\leq_{\mathcal{H}}$. Let $\leqslant$ be $P(\leq)$ and $\leqslant_h$ be $P(\leq_{\mathcal{H}})$. If $\leq$ satisfies $(EE6)$ then $M \leqslant N$ iff $Horn(M) \leqslant_h Horn(N)$ for all logically closed sets M, N.*

Proof. ($\Rightarrow$): Assume $M \leqslant N$, we need to show $Horn(M) \leqslant_h Horn(N)$. It suffices to show for all $\phi \in \mathcal{L}_H$ and $\phi \notin Horn(N)$, there is a $\psi \in \mathcal{L}_H$ and $\psi \notin Horn(M)$ s.t. $\phi \leq_{\mathcal{H}} \psi$. By definition of $Horn(\cdot)$ it is easy to verify that for each $\phi \in \mathcal{L}_H$, $\phi \notin M$ iff $\phi \notin Horn(M)$. It follows from $M \leqslant N$ that for each $\phi \in \mathcal{L}_H$ and $\phi \notin N$ there is a $\psi \notin M$ s.t. $\phi \leq \psi$. As $\leq$ satisfies $(EE6)$ there is $\psi_h \in \psi_{\mathcal{H}}$ s.t. $\psi_h =_{\leq} \psi$. So it follows from $(EE1)$ and $\phi \leq \psi$ that $\phi \leq \psi_h$. Furthermore $\psi_h \notin Horn(M)$ for otherwise $\psi \in M$ as M is closed, contradicting $\psi \notin M$. Finally by Definition 2, we have $\phi \leq_{\mathcal{H}} \psi_h$.
($\Leftarrow$): Assume $Horn(M) \leqslant_h Horn(N)$, we need to show $M \leqslant N$. It suffices to show for all $\phi \notin N$, there is a $\psi \notin M$ s.t. $\phi \leq \psi$. For all $\phi \notin N$, there are two cases to consider:
Case 1: $\phi \in \mathcal{L}_H$. It follows from $Horn(M) \leqslant_h Horn(N)$ that for each such ϕ there is a $\psi \in \mathcal{L}_H$ and $\psi \notin Horn(M)$ s.t. $\phi \leq_{\mathcal{H}} \psi$. As $\psi \notin Horn(M)$ implies $\psi \notin M$, we are done.
Case 2: $\phi \notin \mathcal{L}_H$. As N is closed and $\leq$ satisfies $(EE6)$, there is a $\phi_h \in \phi_{\mathcal{H}}$ s.t. $\phi_h \notin Horn(N)$ and $\phi_h =_{\leq} \phi$. It then follows from $\phi_h \notin Horn(N)$ that there is a $\psi_h \in \mathcal{L}_H$ and $\psi_h \notin Horn(M)$ s.t. $\phi_h \leq_{\mathcal{H}} \psi_h$. By Definition 2 we have $\phi_h \leq \psi_h$ and by $(EE1)$, $\phi \leq \psi_h$. Furthermore as $\psi_h \notin Horn(M)$, $\psi_h \notin M$. We are done. □

Corollary 1. *Given an EE-relation $\leq$ and its Horn subset $\leq_{\mathcal{H}}$. Let $\leqslant$ be $P(\leq)$ and $\leqslant_h$ be $P(\leq_{\mathcal{H}})$. If $\leq$ satisfies $(EE6)$ then the TRPMHC $C(\leqslant_h)$ is Horn equivalent to the TRPMC $C(\leqslant)$.*

Theorem 5 shows that if a P-relation $\leqslant$ is obtained from an EE-relation $\leq$ via Definition 8 then the EEHC based on $\leq$ is identical with the TRPMHC based on $\leqslant$.

Theorem 5. *Let $\leq_{\mathcal{H}}$ be an EE-relation over $\mathcal{L}_H$ (with respect to a belief set H). Then the EEHC $C(\leq_{\mathcal{H}})$ is identical with the TRPMHC $C(P(\leq_{\mathcal{H}}))$.*

Proof. Let $\leq$ be an EE-relation over $\mathcal{L}$ (with respect to the belief set $Cn(H)$) and such that it satisfies $(EE6)$ and $\leq_{\mathcal{H}}$ is its Horn subset (such EE-relation always exists). By Theorem 2, $C(\leq_{\mathcal{H}})$ is Horn equivalent to $C(\leq)$. By Theorem 4 of [1], $C(\leq)$ is identical with $C(P(\leq))$. By Corollary 1, $C(P(\leq_{\mathcal{H}}))$ is Horn equivalent to $C(P(\leq))$. Finally putting these all together we have $C(\leq_H)$ is identical with $C(P(\leq_{\mathcal{H}}))$. □

Clearly, Theorem 5 implies that for any EEHC there is a TRPMHC that performs identically with the EEHC. That is if $\dot{-}_E$ is a EEHC then there is a TRPMHC $\dot{-}_P$ such that $H\dot{-}_E\phi = H\dot{-}_P\phi$ for all H and ϕ.

For the other direction, we start with a P-relation and the TRPMHC it determines. Through Definition 9, an EE-relation is obtained that determines a EEHC. Theorem 6 which follows from Example 1 reveals the connections between the TRPMHC and EEHC.

Theorem 6. *There exists a P-relation $\leqslant$ with respect to a Horn belief set H such that there exist no EE-relations $\leq$ such that $C(\leqslant)$ is identical with $C(\leq)$.*

Theorem 6 suggests that if $\dot{-}_P$ is a TRPMHC then there may not exist a EEHC $\dot{-}_E$ such that $H\dot{-}_P\phi = H\dot{-}_E\phi$ for all H and ϕ.

5 Conclusion

In this paper we have explored the relationship between an account of Horn contraction based on epistemic entrenchment (i.e., EEHC) and Horn contraction based on remainder sets (i.e, TRPMHC). Our investigation shows that turning to Horn logic P-relations as a logical structure for representing an agent's preference information is as effective as its counterpart in propositional logic. In contrast, with Horn EE-relations some preference information is not representable due to the restriction that only Horn formulas are allowed. As a consequence, for each EEHC there is a TRPMHC that performs identically with it but not vice versa. An immediate implication is that the characterising postulates for EEHC are too specific for TRPMHC. In another aspect, our investigation gives a much clearer understanding of the various proposals for Horn contraction allowing a much better comparison between them. According to our results TRPMHC turns out to be more comprehensive than EEHC.

References

1. Rott, H.: Two methods of constructing contractions and revisions of knowledge systems. Journal of Philosophical Logic 20(2), 149–173 (1991)
2. Delgrande, J.P., Wassermann, R.: Horn clause contraction function: Belief set and belief base approaches. In: Proc. KR 2010 (2010)
3. Delgrande, J.P.: Horn clause belief change: Contraction functions. In: Proc. KR 2008, pp. 156–165 (2008)
4. Booth, R., Meyer, T., Varzinczak, I.J.: Next steps in propositional Horn contraction. In: Proc. IJCAI 2009, pp. 702–707 (2009)
5. Booth, R., Meyer, T., Varzinczak, I., Wassermann, R.: A contraction core for horn belief change: Preliminary report. In: Proc. NMR 2010 (2010)
6. Zhuang, Z.Q., Pagnucco, M.: Horn contraction via epistemic entrenchment. In: Proc. JELIA 2010 (2010)
7. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. JSL 50(2), 510–530 (1985)
8. Gärdenfors, P., Makinson, D.: Revisions of knowledge systems using epistemic entrenchment. In: Proc. TARK 1988, pp. 83–95 (1988)
9. Flouris, G., Plexousakis, D., Antoniou, G.: Generalizing the AGM postulates: preliminary results and applications. In: Proc. NMR 2004, pp. 171–179 (2004)
10. McKinsey, J.C.C.: The decision problem for some classes of sentences without quantifiers. JSL 8(2), 61–76 (1943)
11. Dechter, R., Pearl, J.: Structure identification in relational data. Artif. Intell. 58(1-3), 237–270 (1992)
12. Selman, B., Kautz, H.: Knowledge compilation using Horn approximations. In: Proc. AAAI 1991, pp. 904–909. MIT Press, Cambridge (1991)

Database Normalization as a By-product of Minimum Message Length Inference

David L. Dowe and Nayyar Abbas Zaidi

Clayton School of I.T., Monash University, Clayton, Vic. 3800, Australia
{david.dowe,nayyar.zaidi}@infotech.monash.edu.au

Abstract. Database normalization is a central part of database design in which we re-organise the data stored so as to progressively ensure that as few anomalies occur as possible upon insertions, deletions and/or modifications. Successive normalizations of a database to higher normal forms continue to reduce the potential for such anomalies. We show here that database normalization follows as a consequence (or special case, or by-product) of the Minimum Message Length (MML) principle of machine learning and inductive inference. In other words, someone (previously) oblivious to database normalization but well-versed in MML could examine a database and - using MML considerations alone - normalise it, and even discover the notion of attribute inheritance.

Keywords: Minimum Message Length, MML, Database Normalization, Machine Learning, Data Mining, Intelligent Databases.

1 Introduction

The table is a basic building block of a Relational Database Management System (RDBMS) [1, 2]. Consequently, the structure of one or more tables in the database is of great interest. Typically, the information is structured into tables during the Entity-Relationship (ER) diagram phase of conceptual database design. Database normalization [3] is a process of evaluating and correcting table structures to minimize data redundancies, thereby reducing the likelihood of data anomalies upon insertion (adding a new row), deletion (deleting a row) or modification (modifying a row). Normalization is typically the heart of any database design activity.

Database normalization [2] works through a series of stages called normal forms (NFs) (described in sec. 3). A primary reason is to minimize data redundancies and get rid of update, delete and insertion anomalies. Also, designers would like to apply the 'minimal data rule' to the structure, making sure that all information has been captured and every piece of information captured is meaningful. In short, after the initial design is complete in the form of an Entity-Relationship (ER) diagram, designers generally analyze the relationships that exist among attributes within entities through normalization and improve the structure if need arises.

J. Li (Ed.): AI 2010, LNAI 6464, pp. 82–91, 2010.

As described, there can be many motivations behind a database normalization. In this paper, we present a novel information-theoretic perspective of database normalization. We consider the structure of the table(s) as a modelling problem for Minimum Message Length (MML) (see sec. 2). MML seeks a model giving the shortest two-part coding of model and data. If we consider table structure as a model which encodes data, MML advocates that we should be particularly interested in the variation of the encoding length of model and data as the normalization process re-structures tables for efficient design. We will consider a simple example and apply normalization to illustrate our point. As we will see in sec. 4, normalization into higher forms minimizes code length (or message length) by re-structuring the relational tables. Hence, if we apply the MML principle to a relational database, then - provided we have sufficient data - we are likely to get our database normalized.

The rest of the paper is organized as follows: we introduce the minimum message length (MML) framework in sec. 2. In sec. 3, we explain a typical normalization procedure with an example. We explain the MML view of normalization in sec. 4. We conclude in sec. 5.

2 Minimum Message Length

Minimum message length (MML), introduced by Wallace and Boulton [4], is an important stream of studying the complexity of a data set [4–10]. It is based on Shannon's theory of information and equivalently on the theory of Turing machines and Kolmogorov complexity [6, 7, 10]. MML considers any given string S as being a representation in some (unknown) code about the real world. It seeks a ([concatenated] two-part) string $I = H : A$ where the first part H specifies (or encodes) a hypothesis about the data S and the second part A is an encoding of the data using the encoded hypothesis. If the code or hypothesis is true, the encoding is efficient (like Huffman or arithmetic codes). According to Shannon's theory, the length of the string coding an event E in an optimally efficient code is given by $-\log_2(\text{Prob}(E))$, so the length of A is given by:

$$\#A = -\log_2(f(S|H)) \tag{1}$$

where $f(S|H)$ is the conditional probability (or statistical likelihood) of data S given the hypothesis H. Using an optimal code for specification, the length $\#H$ of the first part of the MML message is given by $-\log_2(h(H))$, where $h(\cdot)$ is the prior probability distribution over the set of possible hypotheses. Using equation (1), the total two-part message length $\#I$ is:

$$\begin{aligned} \#I = \#H + \#A &= -\log_2(h(H)) - \log_2(f(S|H)) \\ &= -\log_2(h(H) \times f(S|H)) \end{aligned} \tag{2}$$

The minimization of $\#I$ is equivalent to the maximization of $h(H) \times f(S|H) = \text{Prob}(H, S)$, that is the joint probability of hypothesis and data. It is thus formally equivalent to choosing the hypothesis H of highest Bayesian posterior *probability* (*not* a density) given S [11, secs. 2 and 6.1][8, sec. 11.3.1][9, footnote 158].

3 A Typical Normalization Procedure - An Example

As hinted in sec. 1, normalization works through a series of stages called normal forms. The first three stages are called first normal form (1NF), second normal form (2NF) and third normal form (3NF). From a structural point of view, 2NF is better than 1NF, and 3NF is in turn better than 2NF. For at least many purposes, 3NF is sufficient. A properly designed 3NF structure can also meet the requirements of higher normal forms - such as, e.g., Boyce-Codd Normal Form (BCNF), fourth normal form (4NF) or fifth normal form (5NF).

As an example, suppose we wish to create a database of university student enrolments. Let us assume that a student can take more than one unit, perhaps as many units as she likes. A student can also take the same unit more than once but not in one semester. Each student has a supervisor (denoted as 'lecturer' in the following discussion). A supervisor can have more than one student, but a student can only have one supervisor. Based on this information, we can proceed with the normalization procedure as follows, initially using this small example of $L = 11$ rows in 1NF:

1NF: The term 1NF describes a tabular data format where the following properties hold. First, all of the key attributes are defined. Second, there are no repeating groups in the table -i.e., in other words, each row/column intersection (or cell) contains one and only one value, not a set of values. Third, all attributes are dependent on the primary key (PK). Based on the information given to us, we can structure data in the table Student-Rec and can use Stud-ID, Unit-No and Yr-Sem attributes as parts of the PK. The table in 1NF is shown in table 1.

2NF: A table is in 2NF if the following conditions hold. First, it is in 1NF. Second, it includes no partial dependencies, that is no attribute is dependent on

Table 1. Student-Rec in 1NF. PK = (**Stud-ID**, **Unit-No**, **Yr-Sem**)

Stud-ID	Stud-Name	Stud-Address	Stud-Course	Unit-No	Unit-Name	Lect-No	Lect-Name	Yr-Sem	Grade
212	Bob Smith	Notting Hill	MIT	FIT2014	Database Design	47	Geoff Yu	2007	D
212	Bob Smith	Notting Hill	MIT	FIT3014	Algorithm Theory	47	Geoff Yu	2007	HD
212	Bob Smith	Notting Hill	MIT	EE1007	Circuit Design	47	Geoff Yu	2006	P
213	John News	Caufield	BSc	FIT3014	Algorithm Theory	122	June Matt	2007	HD
213	John News	Caufield	BSc	EE1007	Circuit Design	122	June Matt	2007	HD
214	Alice Neal	Clayton S	BSc	FIT2014	Database Design	122	June Matt	2007	HD
214	Alice Neal	Clayton S	BSc	FIT3014	Algorithm Theory	122	June Matt	2007	D
215	Jill Wong	Caufield	MIT	FIT2014	Database Design	47	Geoff Yu	2007	D
215	Jill Wong	Caufield	MIT	FIT2014	Database Design	47	Geoff Yu	2008	D
216	Ben Ng	Notting Hill	BA	EE1007	Circuit Design	47	June Matt	2007	P
216	Ben Ng	Notting Hill	BA	MT2110	Mathematics-II	47	June Matt	2007	D

Table 2. Student in 2NF. PK = **Stud-ID**

Stud-ID	Stud-Name	Stud-Address	Stud-Course	Lect-No	Lect-Name
212	Bob Smith	Notting Hill	MIT	47	Geoff Yu
213	John News	Caufield	BSc	122	June Matt
214	Alice Neal	Clayton S	BSc	47	Geoff Yu
215	Jill Wong	Caufield	MIT	47	Geoff Yu
216	Ben Ng	Notting Hill	BA	122	June Matt

Table 3. Unit in 2NF and 3NF, PK = **Unit-No**

Unit-No	Unit-Name
FIT2014	Database Design
FIT3014	Algorithm Theory
EE1007	Circuit Design
MT2110	Mathematics-II

Table 4. Stu-Unit-Rec in 2NF and 3NF. PK = (**Stud-ID**, **Unit-No**, **Yr-Sem**)

Stud-ID	Unit-No	Yr-Sem	Grade
212	FIT2014	2007	D
212	FIT3014	2007	HD
212	EE1007	2006	P
213	FIT3014	2007	HD
213	EE1007	2007	HD
214	FIT2014	2007	HD
214	FIT3014	2007	D
215	FIT2014	2007	D
215	FIT2014	2008	D
216	EE1007	2007	P
216	MT2110	2007	D

Table 5. Student in 3NF. PK = **Stud-ID**

Stud-ID	Stud-Name	Stud-Address	Stud-Course	Lect-No
212	Bob Smith	Notting Hill	MIT	47
213	John News	Caufield	BSc	122
214	Alice Neal	Clayton S	BSc	47
215	Jill Wong	Caufield	MIT	47
216	Ben Ng	Notting Hill	BA	122

only a portion of the primary key. The table Student-Rec in table 1 has partial dependencies. It is clear that Unit-Name only depends on Unit-No and not on the whole PK - that is, (Stud-ID, Unit-No,Yr-Sem). Also Name, Address, Course, Lect-No and Lect-Name depend only on Stud-ID. To structure the tables in 2NF, we need to eliminate these partial dependencies. A proposed design modification is shown in tables 2, 3 and 4. Table 1 is split into three tables: Student, Unit and Stu-Unit-Rec. Note that there are no partial dependencies in all of these three tables. In each table, each non-key attribute depends on all attributes in the PK. For our example, the 2NF tables are tables 2, 3 and 4.

3NF: A table is in 3NF if the following holds. First, it is in 2NF. Second, it contains no transitive dependencies. A transitive dependency exists when there are functional dependencies [1] such that $X \rightarrow Y$, $Y \rightarrow Z$ and X is the primary key attribute. Looking at our design in 2NF, there exists a transitive dependency in the Student table (table 2), where Stud-ID $\rightarrow$ Lect-No $\rightarrow$ Lect-Name. This transitive dependency can be eliminated by breaking the Student table into the Student and Lecturer tables as shown in tables 5 and 6. Note the Unit and Stu-Unit-Rec tables in tables 3 and 4 are already in 3NF. For our example, the 3NF tables are tables 3, 4, 5 and 6.

[1] The attribute B is fully functional dependent on the attribute A if each value of A determines one and only value of B.

Table 6. Lecturer in 3NF, PK = **Lect-No**

Lect-ID	Lect-Name
47	Geoff Yu
122	June Matt

4 MML Interpretation of Normalization

Our simple example of the normalization process from the previous section has (ultimately) resulted in four distinct tables - namely, Student (table 5), Lecturer (table 6), Unit (table 3), and Stu-Unit-Rec (table 4). Normalization is nothing but judicious re-structuring of information via tables.

We now flesh out ideas from [9, footnote 187], [12, pp454-455] and [10, sec. 7.6]. In an MML or information-theoretic version of normalization, we can think of tables as a certain model (or hypothesis). Following equation (2), we can write the first-part message length (encoding the model) as:

$$\#H = | < T > | + | < A > | + \sum_{t=1}^{T} AP_t \tag{3}$$

where T is the number of tables, A is the number of attributes, $< T >$ is an encoding of T, $< A >$ is an encoding of A, $| < T > |$ is the length of encoding T and $| < A > |$ is the length of encoding A. AP_t denotes the encoding length of table t's attributes and its primary key. It is defined in equation (4) as:

$$AP_t = \log_2(\mathrm{A}) + \log_2 \binom{A}{a_t} + \log_2(\mathrm{a_t}) + \log_2 \binom{a_t}{p_t} \tag{4}$$

where a_t is the number of attributes in the t^{th} table, p_t denotes the number of attributes in the primary key. (We know that $1 \leq a_t \leq A$, so $\log_2(A)$ is the cost of encoding a_t, and $\log_2 \binom{A}{a_t}$ is the cost of saying which particular a_t attributes are in the t^{th} table. Similarly, since $1 \leq p_t \leq a_t$, $\log_2 a_t$ is the cost of encoding p_t, and $\log_2 \binom{a_t}{p_t}$ is the cost of saying which particular p_t attributes are in the primary key of the t^{th} table.) Note that this is only one way of specifying the model. We have taken only the number of tables, attributes in each table and attributes constituting the PK in each table into account in specifying a model. Other models could be used. Note that the foreign keys (FKs) are not specified in this model - as the model encompasses information about the attributes in each table along with primary keys (PKs), the FKs can be found out by tracking the PK attribute of one table appearing in another table. For the sake of simplicity, we will not consider the effect(s) of $| < T > |$ and $| < A > |$ in the following discussion, as $| < A > |$ appears in the encoding of each normalized form. We could (and implicitly do) assume a uniform prior on $| < T > |$, but we could equally well instead have used (e.g.) a unary code ($\Pr(T) = 2^{-T}$, $| < T > | = T$) or the very slowly growing Wallace tree code [7, fig. 2.13 and sec. 2.1.14]. Hence, neglecting (near-)constant terms, we can (re-)write equation (3) as:

$$\#H = \sum_{t=1}^{T} AP_t \tag{5}$$

In the following discussion we will assume that there are m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9 and m_{10} unique instances of Stud-ID, Stud-Name, Stud-Address, Stud-Course, Unit-No, Unit-Name, Lect-No, Lect-Name, Yr-Sem and Grade respectively (refer to table 7).

Table 7. Number of unique instances for each attribute in table 1, 1NF of our initial example

Stud-ID	Stud-Name	Stud-Address	Stud-Course	**Unit-No**	Unit-Name	Lect-No	Lect-Name	**Yr-Sem**	Grade
m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	m_{10}
5	5	5	5	4	4	2	2	3	3

The number of rows in the 1NF form of the table is an important variable. We have denoted it by L in the preceding equations. $L = 11$ in table 1 and depends on how many students are taking how many courses in each semester. We will later show that there is not a huge need for normalization if each student is taking only one unit, as 2NF will encode the same (amount of) information as 1NF. As more students take more courses, the need for normalization arises.

Let us consider data in the 1NF Student table in table 1. We can write the 1NF encoding length (I_{1NF}) as the sum of the length of model (H_{1NF}) and length of data (A_{1NF}) encoded by this model as follows:

$$\begin{aligned} I_{1NF} &= \#H_{1NF} + \#A_{1NF} \\ &= \#H_{1NF} + L \times (\log_2 m_1 + \log_2 m_2 + \log_2 m_3 + ... + \log_2 m_{10}) \end{aligned} \tag{6}$$

$\#H_{1NF}$ in the preceding equation (equation (6)) can be computed from equations (4) and (5). As there is only one table, $T = 1$. There are 10 attributes ($A = 10$) and 3 attributes in the primary key ($p = 3$).

Consider the three tables used here in 2NF - i.e., Student, Unit and Stu-Unit-Rec (shown in tables 2, 3, 4). We can write the 2NF encoding length (I_{2NF}) as the sum of the length of model (H_{2NF}) and length of data (A_{2NF}) encoded by this model. Examining the 3 tables and their attributes, this gives:

$$\begin{aligned} I_{2NF} &= \#H_{2NF} + \#A_{2NF} \\ &= \#H_{2NF} + m_1 \times (\log_2 m_1 + \log_2 m_2 + \log_2 m_3 + \log_2 m_4 + \log_2 m_7 + \log_2 m_8) \\ &\quad + m_5 \times (\log_2 m_5 + \log_2 m_6) \\ &\quad + L \times (\log_2 m_1 + \log_2 m_5 + \log_2 m_9 + \log_2 m_{10}) \end{aligned} \tag{7}$$

Like $\#H_{1NF}$ (from equation (6)), $\#H_{2NF}$ in the preceding equation (equation (7)) can also be computed from equations (4) and (5). There are 10 attributes ($A = 10$) in total and $T = 3$ tables. The Student table has 6 attributes ($a_1 = 6$) and 1 PK attribute ($p_1 = 1$). Similarly, the Unit table has 2 attributes ($a_2 = 2$) and 1 PK attribute ($p_2 = 1$). The Stu-Unit-Rec table has 4 attributes ($a_3 = 4$)

and 3 PK attributes ($p_3 = 3$). The $\#A_{2NF}$ part in equation (7) is the sum of the encoding lengths of the data in these 3 tables. Note the multiplication factors m_1, m_5 and L in the encoding term, since there are m_1 rows in the Student table, m_5 rows in the Unit table and L rows in the Stu-Unit-Rec table.

Moving from $2NF$ to $3NF$, the Student table in $2NF$ is split into Student (table 5) and Lecturer (table 6). We can write the 3NF encoding length (I_{3NF}) as the sum of the length of model (H_{3NF}) and length of data (A_{3NF}) encoded by this model, noting that we replace the cost of the student table (table 2) in 2NF from equation (7) with the costs of the new (and more concise) student table (table 5) and the lecturer table (table 6).

$$\begin{aligned} I_{3NF} &= \#H_{3NF} + \#A_{3NF} \\ &= \#H_{3NF} + m_1 \times (\log_2 m_1 + \log_2 m_2 + \log_2 m_3 + \log_2 m_4 + \log_2 m_7) \\ &\quad + m_7 \times (\log_2 m_7 + \log_2 m_8) \\ &\quad + m_5 \times (\log_2 m_5 + \log_2 m_6) \\ &\quad + L \times (\log_2 m_1 + \log_2 m_5 + \log_2 m_9 + \log_2 m_{10}) \end{aligned} \tag{8}$$

$\#H_{3NF}$ can also be computed from equations (4) and (5). There are $A = 10$ attributes and $T = 4$ tables in $3NF$. The Student table has 5 attributes ($a_1 = 5$) and 1 PK attribute ($p_1 = 1$). Since the Unit and Stu-Unit-Rec tables are already in $3NF$, we have $a_2 = 2, p_2 = 1, a_3 = 4$ and $p_3 = 3$ from the previous discussion. The Lecturer table has 2 attributes ($a_4 = 2$) and 1 PK attribute ($p_4 = 1$).

The encoding length of data along with the model for each NF for our initial small example (of only $L = 11$ rows in 1NF) is shown in table 8. As we have moved to higher NFs, we have made our model more complicated as depicted by the encoding length ($\#H$), but the data in the second part of the message ($\#A$) is encoded more efficiently and its length has vastly decreased. As can be seen from equations (6), (7) and (8), all encodings depend on the parameter L. We see an improvement of 2NF over 1NF even for $L = 11$ rows in this small example.

Table 8. Code length (bits) of model and data for different NFs on small example

	$\#H$ (first part's length)	$\#A$ (second part's length)	total message length
1NF	10.22	203.03	213.25
2NF	36.45	154.89	191.34
3NF	46.26	153.84	200.10

Due to space constraints we have not included a lot of data in table 1. In this particular example with $L = 11$ rows in 1NF, the total message length appears (slightly) higher for 3NF than 2NF. This should not be surprising considering the amount of data we have. Let us note from equations (7) and (8) that

$$I_{3NF} - I_{2NF} \quad = \quad (\#H_{3NF} - \#H_{2NF}) \; + \; m_7 \log_2 m_7 \; + \; (m_7 - m_1) \log_2 m_8$$

On the not unreasonable assumption that $m_7 = m_8$, then $I_{3NF} - I_{2NF} = (\#H_{3NF} - \#H_{2NF}) + (2m_7 - m_1)\log_2 m_7$.

Provided that $m_1 > 2m_7$ then, as m_1 or m_7 increases, the term $(2m_7 - m_1)\log_2 m_7$ will become increasingly negative, eventually becoming larger in magnitude than $(\#H_{3NF} - \#H_{2NF})$, whereupon I_{3NF} will become less than I_{2NF}, at which point MML will then forever after prefer 3NF to 2NF. This comparison between m_1 and $2m_7$ is because in going from 2NF to 3NF we are removing a column of m_1 entries in the 2NF Student table (table 2) and replacing it with a new 3NF table (table 6, Lecturer) of 2 columns and m_7 rows.

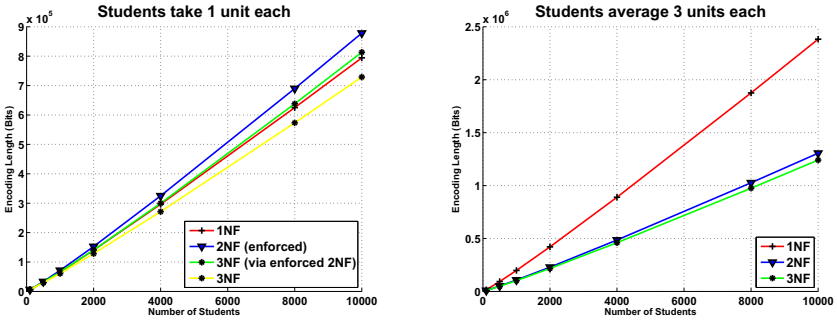
So, now let us suppose that we have a more realistic (but still quite small) example of $m_1 = 100$ students, $m_5 = 30$ units and each student is taking an average of 3 courses (note $L = 300$), setting the number of lecturers equal to $m_7 = 15$. The encoding lengths are given in table 9, which is also a cross-section of figure 1(b).

Table 9. Encoding length (in bits) of model and data for different NFs, Number of Students (m_1) = 100, Number of Units (m_5) = 30, Number of Lecturers (m_7) = 15, $L = 300$

	$\#H$ (first part's length)	$\#A$ (second part's length)	total message length
1NF	10.22	14210	14220
2NF	36.45	8150	8186
3NF	46.26	7876	7922

To illustrate this point graphically, in figure 1 we see the effect on encoding length by varying and increasing L and the number of students (m_1). If each student is only taking one unit ($m_1 = m_5$), 2NF will not be beneficial even if the number of students is increased from 10 to say 10000. This is depicted in figure 1(a). Because $L = m_1$ and there is insufficient data to infer the partial dependencies required for 2NF, the original 1NF table is adequate for 2NF. Indeed, enforcing the premature creation of (superfluous) tables (enforced 2NF) can understandably be seen to increase the message length. Despite this, the transitive dependencies of Stud-ID → Lect-No → Lect-Name (with $m_1 > 2m_7$) result in message length improvements when we go to 3NF. Taking the unnecessarily enforced 2NF is improved by then converting it to 3NF. But best of all is to take the original 1NF table as our 2NF (as there is insufficient data to suggest otherwise) and then convert this to 3NF. Figure 1(a) bears out this analysis.

As can be seen from figure 1(b) (of which the total message length in table 9 is a special case with $m_1 = 100$ and $L = 3m_1 = 300$), normalization from 1NF to 2NF is really beneficial as students enrol in more than one unit. The emphatic message length benefits visible in figure 1(b) in going from 1NF to 2NF are most probably to be expected, with the pronounced benefit of normalization being self-evident when the number of students (enrolling in more than one unit) is large. The transitive dependency that we observed in figure 1(a) applies again



(a) (b)

Fig. 1. Variation in total message length (I) by varying number of students (m_1) and L for different NFs. The number of Units (m_5) is set to 30 and the number of Lecturers (m_7) is set to 15. 1(a): $L = m_1$, 1(b): $L = 3m_1$.

here (with $m_1 \geq 50 > 2m_7 = 30$) as the number of students (m_1) and their enrolments increases, whereupon MML again prefers the 3NF model.

5 Conclusion and Future Work

We have presented database normalization as a consequence of MML inference. With an example, we demonstrated a typical normalization procedure and analyzed the process using the MML framework. We found that with higher NFs, the model is likely to become more complicated, but the data encoding length is decreased. If there is a relationship or dependency in the data (according to database normalisation principles), then - given sufficient data - MML will find this. This suggests that normalization is - in some sense - simply following MML.

Though we have limited ourselves here to 1st, 2nd and 3rd normal forms (NFs), applying MML can also be shown to lead to higher NFs such as Boyce-Codd Normal Form (BCNF), 4NF and 5NF. Indeed, recalling the notion of MML Bayesian network (see, e.g., [7, sec. 7.4][8][9, sec. 0.2.5 and footnotes 53, 75 & 218][10]), normalizing and breaking down tables into new tables can be thought of as a (MML) Bayesian net analysis - using the fact that (in some sense) databases could be said to have no noise. And, in similar manner, (the notion of) attribute inheritance (where different types of employee - such as pilot and engineer - have their own specific attributes as well as inheriting common employee attributes) can also be inferred using MML. General statistical consistency results (see, e.g., [8, sec. 11.3][9, secs. 0.2.3 - 0.3.1][12, pp436-437][10, sec. 5.2]) appear to guarantee that - given sufficiently large data-sets and sufficient search time - MML will converge upon the data generation process, whatever it is, whatever the appropriate (possibly previously unseen) normalization (or regularities). Our initial results here provide clear supporting evidence of this claim.

References

1. Codd, E.: A relational model of data for large shared data banks. Communications of the ACM (1979)
2. Date, C.: An Introduction to Database Systems. Addison-Wesley Longman, Amsterdam (1999)
3. William, K.: A simple guide to five normal forms in relational database theory. Communications of the ACM (1983)
4. Wallace, C.S., Boulton, D.M.: An information measure for classification. Computer Journal 11, 185–194 (1968)
5. Wallace, C.S., Freeman, P.R.: Estimation and inference by compact coding. Journal of the Royal Statistical Society series B 49(3), 240–252 (1987)
6. Wallace, C.S., Dowe, D.L.: Minimum message length and Kolmogorov complexity. Computer Journal 42(4), 270–283 (1999)
7. Wallace, C.S.: Statistical and Inductive Inference by Minimum Message Length. Information Science and Statistics. Springer, Heidelberg (May 2005)
8. Comley, J.W., Dowe, D.L.: Minimum message length and generalized Bayesian nets with asymmetric languages. In: Grünwald, P., Pitt, M.A., Myung, I.J. (eds.) Advances in Minimum Description Length: Theory and Applications, pp. 265–294. M.I.T. Press, Cambridge (April 2005)
9. Dowe, D.L.: Foreword re C. S. Wallace. Computer Journal 51(5), 523–560 (2008); Christopher Stewart WALLACE (1933-2004) memorial special issue
10. Dowe, D.L.: MML, hybrid Bayesian network graphical models, statistical consistency, invariance and uniqueness. In: Wood, J., Forster, M.R., Bandyopadhyay, P. (eds.) Handbook of the Philosophy of Science - (HPS Volume 7) Philosophy of Statistics, vol. 7, pp. 861–942. Elsevier, Amsterdam (2010)
11. Wallace, C.S., Dowe, D.L.: MML clustering of multi-state, Poisson, von Mises circular and Gaussian distributions. Statistics and Computing 10, 73–83 (2000)
12. Dowe, D.L.: Minimum Message Length and statistically consistent invariant (objective?) Bayesian probabilistic inference - from (medical) "evidence". Social Epistemology 22(4), 433–460 (2008)

A Heuristic on Effective and Efficient Clustering on Uncertain Objects

Edward Hung, Lei Xu, and Chi-Cheong Szeto

Department of Computing, The Hong Kong Polytechnic University
{csehung,cslxu,csccszeto}@comp.polyu.edu.hk

Abstract. We study the problem of clustering uncertain objects whose locations are uncertain and described by probability density functions. We analyze existing pruning algorithms and experimentally show that there exists a new bottleneck in the performance due to the overhead while pruning candidate clusters for assignment of each uncertain object in each iteration. We further show that by considering squared Euclidean distance, UK-means (without pruning techniques) is reduced to K-means and performs much faster than pruning algorithms, however, with some discrepancies in the clustering results due to the different distance functions used. Thus, we propose Approximate UK-means to heuristically identify objects of boundary cases and re-assign them to better clusters. Our experimental results show that on average the execution time of Approximate UK-means is only 25% more than K-means and our approach reduces the discrepancies of K-means' clustering results by more than 70% at most.

Keywords: clustering, uncertain objects, UK-means.

1 Introduction

In this paper, we consider the problem of clustering objects with multi-dimensional uncertainty where an object is represented by an uncertain region over which a discrete probability distribution function (PDF) or a probability density function (pdf) is defined. Formally, we consider a set of n objects o_i, $1 \leq i \leq n$ in an m-dimensional space. An object o_i is represented by a pdf f_i: $IR^m \rightarrow IR$ (IR represents real number space) that specifies the probability density of each possible location of object o_i. The goal of clustering is to group these objects into K clusters so that the sum of *expected Euclidean distances* (EED) [2] between the uncertain objects and their cluster centers is minimized. Thus, suppose $C(o_i) = c_j$ represents that object o_i is assigned to cluster c_j, and $p_{C(o_i)}$ is the cluster's representative point, we want to find the K cluster representatives such that the objective function $\sum_{i=1}^{n} EED(o_i, p_{C(o_i)}) = \sum_{i=1}^{n}(\int f_i(x)ED(x, p_{C(o_i)})dx)$ is minimized where ED is the Euclidean distance function.

UK-means [2] is a generalization of the traditional K-means algorithm to handle uncertain objects whose locations are represented by pdfs. For arbitrary pdfs, the bottleneck of UK-means is the calculations of expected distance, which

J. Li (Ed.): AI 2010, LNAI 6464, pp. 92–101, 2010.

are computationally expensive. Our contributions of this paper include (i) after applying the analytic solution in [12] to reduce UK-means to K-means, we experimentally show that K-means performs much faster than existing pruning algorithms proposed in [9,5] with some discrepancies in the clustering results due to the different distance functions used; and (ii) we propose Approximate UK-means to heuristically identify objects of boundary cases and re-assign them to better clusters. Our experimental results show that on average the execution time of Approximate UK-means is only 25% more than K-means (while pruning algorithms are 300% more) and our approach reduces the discrepancies of K-means' clustering results up to 70% .

The rest of the paper is organized as follows. Section 2 briefly describes related work. In Section 3, we introduce expected squared Euclidean distance, and propose a heuristic Approximate UK-means algorithm. Section 4 demonstrates the efficiency and effectiveness of our method by extensive experiments. Finally, Section 5 concludes the paper.

2 Related Work

Pruning techniques were proposed to improve the efficiency of UK-means by reducing unnecessary expected distance calculations. In MinMax-BB [9], each object o_i has a minimum bounding rectangle (MBR)[1] outside which the object has zero (or negligible) probability of occurrence. The minimum distance ($MinDist_{i,j}$) and the maximum distance ($MaxDist_{i,j}$) are calculated to prune unnecessary expected distance calculations. The overhead of MinMax-BB includes the time of $MinDist_{i,j}$ and $MaxDist_{i,j}$ calculations. VDBi [5] is another pruning method using Voronoi diagrams [3] to consider the spatial relationships among cluster representatives and is more efficient than MinMax-BB by using Voronoi-cell pruning and bisector pruning. The overhead of VDBi includes the time of Voronoi diagrams construction, Voronoi-cell pruning and bisector pruning. The pruning methods can be more efficient with the use of cluster-shift technique. The additional overhead of SHIFT technique includes the time of cluster representative shift calculation between two consecutive iterations. Although the pruning techniques have reduced most of expected distance calculations, it is still expensive to use these pruning techniques for each object in each iteration. Thus, the pruning process becomes a new bottleneck.

Recently, there have been studies on density-based clustering of uncertain data. FDBSCAN [6] and FOPTICS [7] are based on DBSCAN [4] and OPTICS [1] respectively to handle density-based clustering of uncertain objects. A related area of research is fuzzy clustering [10]. While their work focuses on creating fuzzy clusters, our work is developed for clustering based on the uncertain model of objects, in which each object is uncertain about its location but each object must belong to one certain cluster.

[1] The pruning techniques in [9,5] require that for each object o_i, the uncertain region A_i of each object o_i is finite, i.e. $\forall x \notin A_i, f_i(x) = 0$. Thus, each object can be bounded by a finite bounding box.

3 Clustering Uncertain Objects with Expected Squared Euclidean Distance

The bottlenecks of uncertain object clustering are expected distance calculation and pruning of candidate clusters. However, by considering squared Euclidean distance (instead of Euclidean distance as in UK-means), UK-means can be reduced to K-means (so, no pruning of clusters is necessary) [8], which is running much faster with some discrepancies in the clustering results as shown in the experimental section. In Section 3.2, we will introduce our "Approximate UK-means" which heuristically identifies objects on the boundary cases and reassigns them to better clusters in order to reduce the discrepancies in clustering results.

3.1 Reduce UK-means to K-means

An uncertain object can also be represented as a set of points, each of which is a possible location of object o_i [9,12,5]. As Figure 1(a) shows, the uncertain domain is divided into a number of grid cells. Each grid cell represents a possible location of the uncertain object o_i. The expected Euclidean distance (EED) from object o_i (represented by a pdf f_i) to the cluster representative p_{c_j} is the weighted average of the distances between the samples in o_i and p_{c_j}, i.e. $EED(o_i, p_{c_j}) = \sum_{t=1}^{T} F_i(s_{i,t}) ED(s_{i,t}, p_{c_j})$, where T is the number of samples in o_i, $s_{i,t}$ is the location (vector) of the tth sample of o_i, p_{c_j} is the location (vector) of representative of cluster c_j, F_i is a discrete probability distribution function over T grid cells, and the metric ED is Euclidean distance used in [9,5,2].

By using expected squared Euclidean distance, [8] shows that UK-means algorithm can be reduced to K-means. In the following, we are going to show another derivation by applying the analytic solution in [12]. We will first define the mean vector and the trace of covariance matrix of an uncertain object given its samples as follows. Suppose $\overline{s_i}$ is a $m \times 1$(m is the number of dimensions) mean vector of an uncertain object o_i, which is the weighted mean of all T samples (or possible locations) in the object. Formally, $\overline{s_i} = \sum_{t=1}^{T} s_{i,t} \times F_i(s_{i,t})$. Suppose Σ_i is a $m \times m$ covariance matrix of samples of object o_i. $trace(\Sigma_i)$ is the sum of all diagonal elements in Σ_i. On the other hand, $trace(\Sigma_i)$ can also be expressed as: $trace(\Sigma_i) = \sum_{t=1}^{T} ||s_{i,t} - \overline{s_i}||^2 \times F_i(s_{i,t})$, where $s_{i,t}$ represents the tth sample of object o_i. In [12], the expected squared Euclidean distance (ESED) between two uncertain objects o_i and o_j can be obtained by $ESED(o_i, o_j) = ||\overline{s_i} - \overline{s_j}||^2 + trace(\Sigma_i) + trace(\Sigma_j)$. Consider ESED between object o_i and a cluster representative p_{c_j}. Since p_{c_j} is a certain point, so $\overline{s_j} = p_{c_j}$ and $trace(\Sigma_j) = 0$, and we can obtain $ESED(o_i, p_{c_j}) = ||\overline{s_i} - p_{c_j}||^2 + trace(\Sigma_i)$.

It is obvious that we can preprocess the uncertain objects and obtain their $\overline{s_i}$ and $trace(\Sigma_i)$ in the beginning so that ESED between any object o_i and any cluster representative p_{c_j} can be readily obtained. Given an uncertain object o_i, to find the closer one out of two cluster representatives p_{c_j} and p_{c_k}, we

could calculate the difference between their ESED from o_i: $ESED(o_i, p_{c_k}) - ESED(o_i, p_{c_j}) = (||\overline{s_i} - p_{c_k}||^2 + trace(\Sigma_i)) - (||\overline{s_i} - p_{c_j}||^2 + trace(\Sigma_i)) = ||\overline{s_i} - p_{c_k}||^2 - ||\overline{s_i} - p_{c_j}||^2$. As a result, it is no longer necessary to calculate $trace(\Sigma_i)$ of uncertain objects. Instead of calculating ESED, we only need to calculate the Means' Squared Euclidean distance (MSED) between uncertain object o_i and cluster representative p_{c_j} as follows: $MSED(o_i, p_{c_j}) = ||\overline{s_i} - p_{c_j}||^2$. Moreover, in our algorithm, cluster representatives p_{c_j} are obtained by $p_{c_j} = \frac{1}{|c_j|} \sum_{i=1}^{|c_j|} \overline{s_i}$, where $|c_j|$ is the number of objects assigned to cluster c_j.

3.2 Approximate UK-means

K-means uses squared Euclidean distance while UK-means uses Euclidean distance. It is not surprising that the clustering results of K-means will deviate from those of UK-means. Figure 1(b) gives an example of discrepancy between different distance functions. Assume an uncertain object has two samples in two grid cells and the probabilities of these two samples are equal. The uncertain object is assigned to cluster c_1 if the expected distance is used ($\frac{1+5}{2} < \frac{3+4}{2}$). However, it is assigned to cluster c_2 if the expected squared distance is used instead ($\frac{1^2+5^2}{2} > \frac{3^2+4^2}{2}$). Otherwise, the orders of expected distances between each object and clusters are the same as those of expected squared distances. Figure 1(c) gives an example of the same clustering result. The assumption of this case is the same as that of Figure 1(b). The uncertain object is assigned to cluster c_1 if the expected distance is used ($\frac{1+2}{2} < \frac{3+4}{2}$), and it is also assigned to cluster c_1 if the expected squared distance is used instead ($\frac{1^2+2^2}{2} < \frac{3^2+4^2}{2}$).

Definition of Boundary Case. p_{c_m} and p_{c_q} are two closest cluster representatives of object o_i. $average_{MSED}$ (Means' Squared Euclidean Distance) is the average of $MSED(o_i, p_{c_m})$ and $MSED(o_i, p_{c_q})$. Assume $MSED(o_i, p_{c_m}) < MSED(o_i, p_{c_q})$, if $MSED(o_i, p_{c_q}) \leq \beta \times average_{MSED}$, it means that the two MSED from the two closest cluster representatives are so close and o_i is on the boundary of cluster c_m and c_q. We notice that objects assigned to a cluster that is different from another algorithm are likely on the boundary of clusters. Therefore, we propose a heuristic called Approximate UK-means. The basic idea of the heuristic is picking out boundary objects o_i and re-assigning them in the first iteration. Algorithm 1 shows the algorithm, where n is the number of objects, and K is the number of clusters. If object o_i is boundary case, we calculate the expected Euclidean distances from o_i to p_{c_m} and p_{c_q} and assign o_i to the closest cluster. **The above is only done in the first iteration because we observed that the assignment of objects in the first iteration is the most important, which will greatly affect the later iterations due to the shift of cluster representatives.** In the heuristic, we only consider the uncertain objects that are on the boundaries of two clusters. Another factor β affects the improvement in clustering result discrepancies by identifying boundary cases. If the value of β can pick out boundary cases as many as possible, the discrepancies can be reduced a lot. In experiments, we set β from 0.7 to 1. When β is 1, the algorithm is reduced to K-means. In fact, experimental results in the

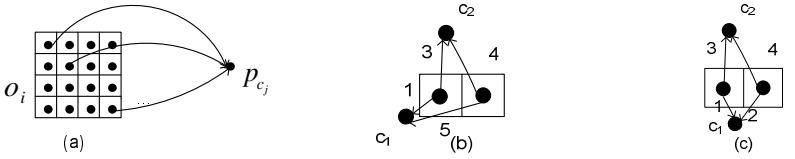


Fig. 1. (a) Expected distance calculation from o_i to p_{c_j} in [9,5,2] (b) An example of the cause of discrepancy (c) An example of the same clustering result

Algorithm 1. Approximate UK-means

randomly initialize all cluster representatives (reps) p_{c_j};
for i=0; $i < n$; i++ **do**
  precompute the mean vector $\overline{s_i}$ of object o_i;
end for
repeat
  for i=0; $i < n$; i++ **do**
    for j=0; $j < K$; j++ **do**
      compute Means' Squared Euclidean Distance $MSED(o_i, p_{c_j}) = ||\overline{s_i} - p_{c_j}||^2$;
    end for
    if this is the first iteration **then**
      let p_{c_m} and p_{c_q} be the 1st and 2nd closest cluster reps by $MSED$;
      $average_{MSED} := (MSED(o_i, p_{c_m}) + MSED(o_i, p_{c_q}))/2$;
      if $MSED(o_i, p_{c_m}) \leq \beta \times average_{MSED}$ **then**
        assign object o_i to cluster c_m;
      else
        compute o_i's expected Euclidean distances from p_{c_m} and p_{c_q}, and assign o_i to cluster with smaller EED;
      end if
    else
      assign object o_i to cluster c_m where p_{c_m} is the closest cluster rep by $MSED$;
    end if
  end for
  update all cluster reps p_{c_j} by $p_{c_j} = \frac{1}{|c_j|} \sum_{i=1}^{|c_j|} \overline{s_i}$;
until all cluster reps converge

next section show that this heuristic can significantly reduce the discrepancies of clustering results by 70% at most compared with K-means with only 25% more execution time.

4 Experimental Evaluation

In this section, we evaluate Approximate UK-means experimentally by comparing it with K-means and pruning UK-means (MinMax-SHIFT and VDBi-SHIFT). Section 4.1 compares their execution time and Section 4.2 compares their clustering results. All algorithms were written in Java 1.5 and were run on a Linux machine with an Intel 2.5GHz Pentium(R) Dual-Core processor and 8GB of main memory.

Table 1. Parameters for experiments using random data sets

Parameter	Description	Baseline Value
n	number of uncertain objects	20000
K	number of clusters	50
T	number of samples per object	196
S	maximum size of MBR, $S \times S$	5
mindis	minimum distance between two clusters	2
D	number of dimensions	2
σ	standard deviation of Gaussian distribution	16

4.1 Execution Time

Random data sets and Gaussian data sets: For ease of comparison with previous work like [5,9,11] which used synthetic data sets only, we generated 125 random data sets for the experiments. For each data set, a set of n uncertain objects represented by MBRs with size 5×5 was randomly generated in 2D space $[0, 100] \times [0, 100]$. An MBR is divided into $\sqrt{T} \times \sqrt{T}$ grid cells. Each grid cell corresponds to a sample. Each sample is associated with a randomly generated probability value. All probabilities in an MBR are normalized to have their sum equal to 1. For each data set, a set of K cluster representatives was randomly initialized and was repeatedly used in all experiments on the same data set. This is to eliminate variations in the results due to the uses of different sets of initial cluster representatives. To make the clustering results more reasonable, we also generated 125 data sets with Gaussian distribution. The n uncertain objects in a data set were equally grouped into K clusters. For each cluster, the centers of $\frac{n}{K}$ uncertain objects were generated from a Gaussian distribution, whose mean and standard deviation equal to the cluster center and σ respectively. The cluster center was randomly generated and was restricted to have a minimum distance *mindis* with other cluster centers. The parameters used for the experiments are summarized in Table 1. For each set of parameters, a set of five experiments was run on five different randomly generated data sets. Each experiment was repeated on the four algorithms. The average value of 5 runs on each algorithm was taken and reported.

Varying Sample Number: We varied the sample number T per object from 100 to 900. The other parameters were kept at baseline values. Figure 2 shows the execution time of the four algorithms on random data sets. Figure 2(a) shows that the total execution time of all four algorithms increases as sample number T increases on random data sets. However, Approximate UK-means runs almost as fast as K-means and its execution time grows much slower than MinMax-SHIFT and VDBi-SHIFT. The significant improvement in the performance of Approximate UK-means is due to two reasons: (i) the distance calculations are done much faster (Figure 2(b)), and (ii) the overhead is much reduced as no pruning is necessary (Figure 2(c)). Figure 2(b) also shows that the (expected) distance calculation time does not change a lot with sample number because MSED calculation does not depend on sample number, and the expected distance

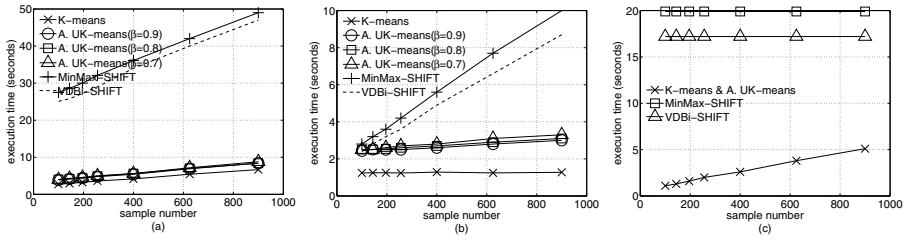


Fig. 2. (a) Total clustering time with varying T on random data sets (RDS) (b) (Expected) distance calculation time with varying T on RDS (c) Overhead time with varying T on RDS

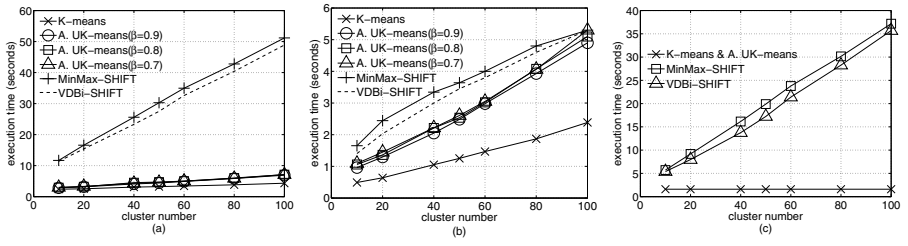


Fig. 3. (a) Total clustering time with varying K on RDS (b) (Expected) distance calculation time with varying K on RDS (c) Overhead time with varying K on RDS

calculation in the first iteration is only a minor cost. The case of Gaussian data sets with varying T is similar to Figure 2. The execution time with varying n and S on random and Gaussian data sets is much similar to that of varying T on both data sets, so we do not show it here because of page limitation.

Varying Cluster Number: We varied the cluster number K from 10 to 100. The other parameters were kept at baseline values. Figure 3 shows the execution time of the four algorithms on random data sets. Figure 3(a) shows that the total execution time of all four algorithms grows as K increases. However, Approximate UK-means almost spends the same time as K-means and its execution time grows much slower than MinMax-SHIFT and VDBi-SHIFT. The distance calculation in Approximate UK-means is more efficient than that in pruning techniques (Figure 3(b)). The overhead time of Approximate UK-means is not related to K while the overhead time of pruning techniques grows linearly with K (Figure 3(c)). The case of Gaussian data sets with varying K is similar to Figure 3, so we do not show it here because of page limitation.

Varying Dimension Number: We varied the dimension number D from 2 to 6 on random data sets. The other parameters were kept at baseline values. Figure 4 shows the execution time of the four algorithms on random data sets. Figure 4(a) shows that the total execution time of all algorithms increases as D increases. Approximate UK-means runs almost as fast as K-means and its execution time grows much slower than MinMax-SHIFT and VDBi-SHIFT. The

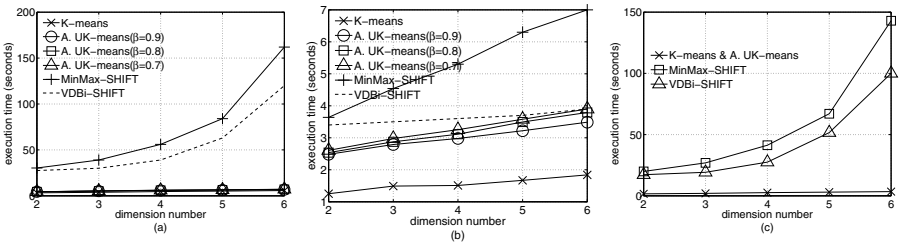


Fig. 4. (a) Total clustering time with varying D on RDS (b) (Expected) distance calculation time with varying D on RDS (c) Overhead time with varying D on RDS

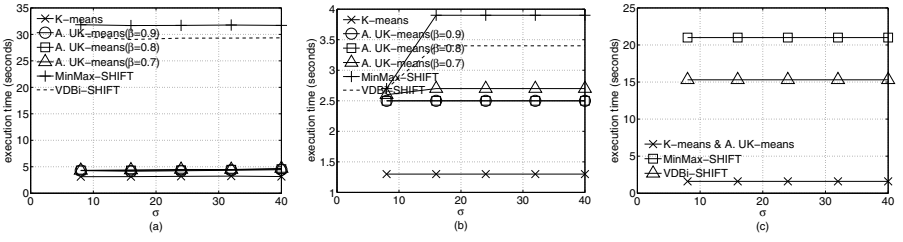


Fig. 5. (a) Total clustering time with varying σ on Gaussian data sets (GDS) (b) (Expected) distance calculation time with varying σ on GDS (c) Overhead time with varying σ on GDS

significant improvement in the performance of Approximate UK-means is due to two reasons: (i) the distance calculations are done much faster in low dimension space (Figure 4(b)), and (ii) the overhead is much reduced as no pruning is necessary (Figure 4(c)).

Varying σ: We varied the standard deviation σ per cluster from 8 to 40 on Gaussian data sets. The other parameters were kept at baseline values. Figure 5 shows the execution time of the four algorithms on Gaussian data sets. Figure 5(a) shows that the total execution time of all algorithms does not increase as σ increases, and the execution time in Approximate UK-means and K-means is much fast than that of MinMax-SHIFT and VDBi-SHIFT, because Figure 5(b) and (c) show that the time of distance calculation and overhead in Approximate UK-means and K-means is much faster then that of pruning techniques. **In Figures(c), the overhead time used for pruning in MinMax-SHIFT and VDBi-SHIFT occupies a large part of total execution time, which is the new bottleneck of pruning UK-means, and the overhead time in Approximate UK-means are all the same for different β values.**

4.2 Clustering Results

Discrepancy is used to measure the difference of clustering results of clustering algorithm A with respect to another clustering algorithm B based on purity. Purity is the maximum probability that a cluster in algorithm A contains objects

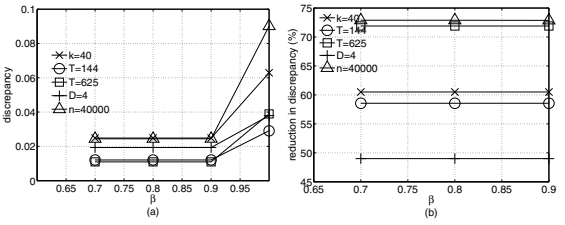


Fig. 6. (a) Discrepancy of Approximate UK-means and K-means (b) reduction in discrepancy of clustering results of the Approximate UK-means on RDS as β varies

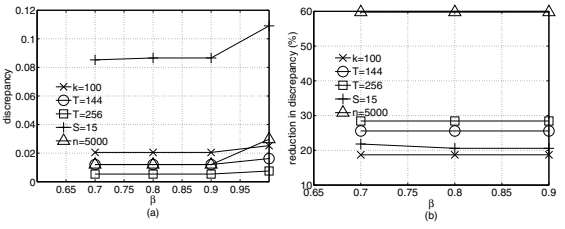


Fig. 7. (a) Discrepancy of Approximate UK-means and K-means (b) reduction in discrepancy of clustering results of the Approximate UK-means on GDS as β varies

of the same cluster from algorithm B. The purity of cluster c_i is defined as $prob_i = \max_j prob_{ij} = \max_j \frac{n_{ij}}{n_i}$ where $prob_{ij}$ is the probability that a member of cluster c_i in algorithm A belongs to cluster c_j in algorithm B, n_i is the number of objects in cluster c_i in algorithm A, and n_{ij} is the number of objects of cluster c_j (in algorithm B) within these n_i objects. The overall purity of clustering result of algorithm A is described as $purity = \sum_{i=1}^{K} \frac{n_i}{n} prob_i$, where K is the number of clusters, n is the number of uncertain objects, and the range of purity is $[0, 1]$. Finally we define $discrepancy = 1 - purity$. **It is emphasized that there is no "true" or "correct" clustering result because even the K-means running on traditional certain objects may give different clustering results by using different distance metrics. In fact, the discrepancy between the results of UK-means and K-means is due to the different distance metrics used. It does not mean that the clustering result by K-means is *wrong*. Therefore, here we would like to only point out the changes that K-means may bring to UK-means (given the same seeds) and also how much Approximate UK-means may reduce these changes.** For each comparison, we ran Approximate UK-means with different β values ranging from 0.7 to 1 to study the effect of β on the discrepancy. Note that when β equals 1, Approximate UK-means is reduced to K-means. We show effect of Approximate UK-means with varying different parameters on some values in Figure 6 and Figure 7. Figure 6(a) and Figure 7(a) show the discrepancy between the clustering results of Approximate UK-means and UK-means with respect to pruning UK-means. Figure 6(b) and Figure 7(b) show the

reduction in discrepancy of Approximate UK-means with respect to K-means. The figures show that the reduction is stable as β decreases. We can see from the Figures that the discrepancy can be reduced by Approximate UK-means, and the reduction can be up to 70%.

5 Conclusion

In this paper, we studied the problem of clustering objects whose locations are presented by probability density functions (pdf). Our experiments show that K-means by using squared Euclidean distance is much faster than UK-means which generates cluster results with discrepancies from the original clustering results of UK-means, which can be reduced by Approximate UK-means.

Acknowledgement

The work described in this paper was partially supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 5191/09E, PolyU 5182/08E, PolyU 5181/06E, PolyU 5174/07E, PolyU A-PH40).

References

1. Ankerst, M., Breunig, M.M., Kriegel, H.-P., Sander, J.: Optics: Ordering points to identify the clustering structure. In: SIGMOD Conference, pp. 49–60 (1999)
2. Chau, M., Cheng, R., Kao, B., Ng, J.: Uncertain data mining: An example in clustering location data. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) PAKDD 2006. LNCS (LNAI), vol. 3918, pp. 199–204. Springer, Heidelberg (2006)
3. Dehne, F.K.H.A., Noltemeier, H.: Voronoi trees and clustering problems. Inf. Syst. 12(2), 171–175 (1987)
4. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.W.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, pp. 226–231 (1996)
5. Kao, B., Lee, S.D., Cheung, D.W., Ho, W.-S., Chan, K.F.: Clustering uncertain data using voronoi diagrams. In: ICDM, pp. 333–342 (2008)
6. Kriegel, H.-P., Pfeifle, M.: Density-based clustering of uncertain data. In: KDD, pp. 672–677 (2005)
7. Kriegel, H.-P., Pfeifle, M.: Hierarchical density-based clustering of uncertain data. In: ICDM, pp. 689–692 (2005)
8. Lee, S.D., Kao, B., Cheng, R.: Reducing uk-means to k-means. In: ICDM Workshops, pp. 483–488 (2007)
9. Ngai, W.K., Kao, B., Chui, C.K., Cheng, R., Chau, M., Yip, K.Y.: Efficient clustering of uncertain data. In: ICDM, pp. 436–445 (2006)
10. Ruspini, E.H.: A new approach to clustering. Information and Control 15(1), 22–32 (1969)
11. Tsang, S., Kao, B., Yip, K.Y., Ho, W.-S., Lee, S.D.: Decision trees for uncertain data. In: ICDE, pp. 441–444 (2009)
12. Xiao, L.R., Hung, E.: An efficient distance calculation method for uncertain objects. In: CIDM, pp. 10–17 (2007)

A New Supervised Term Ranking Method for Text Categorization

Musa Mammadov, John Yearwood, and Lei Zhao

Graduate School of information Technology and Mathematical Science,
University of Ballarat, Ballarat, VIC, 3350, Australia
{m.mammadov,j.yearwood,l.zhao}@ballarat.edu.au

Abstract. In text categorization, different supervised term weighting methods have been applied to improve classification performance by weighting terms with respect to different categories, for example, Information Gain, χ^2 statistic, and Odds Ratio. From the literature there are three term ranking methods to summarize term weights of different categories for multi-class text categorization. They are Summation, Average, and Maximum methods. In this paper we present a new term ranking method to summarize term weights, i.e. Maximum Gap. Using two different methods of information gain and χ^2 statistic, we setup controlled experiments for different term ranking methods. Reuter-21578 text corpus is used as the dataset. Two popular classification algorithms SVM and Boostexter are adopted to evaluate the performance of different term ranking methods. Experimental results show that the new term ranking method performs better.

1 Introduction

The task of text categorization is to assign unlabelled documents to predefined categories (topics or themes) according to their contents. Due to the growth in the volume of electronic documents, text categorization has been widely researched and applied in organizing as well as in finding information on the huge electronic resources.

Term weighting is an important issue for text categorization. In recent years, we have witnessed an increasing number of term weighting methods published. [1] classify the term weighting methods into two categories according to whether the method makes use of known information on the membership of training documents or not, namely, *supervised term weighting methods* and *unsupervised term weighting methods*. For example, *tf* and *tf-idf* [2] weighting methods belong to the unsupervised term weighting methods; information gain, χ^2 statistic, and odds ratio [3,4] are classified as supervised term weighting methods.

Although different approaches have been explored [4], not much attention has been paid towards specific class-oriented and local, context-dependent filters [5]. In particular, for multi-class text categorization, after being weighted by certain weighting methods, for example, information gain, term weights need to be summarized in to a single weight according to different categories.

J. Li (Ed.): AI 2010, LNAI 6464, pp. 102–111, 2010.

The literature indicates that there are only three methods to summarize term weights. The most common method is called the Summation method [3,6], which sums up all terms' weights with respect to all categories. We denote this method by $f_{sum}(t_i) = \sum_{k=1}^{c} f(t_i, c_k)$, where $f(t_i, c_k)$ denotes term t_i's weight with respect to category c_k. [4] employ two other methods. One is the average weight of term t_i with respect to all categories, denoted by $f_{avg}(t_i) = \sum_{k=1}^{c} P(c_k) f(t_i, c_k)$, where $P(c_k)$ is the proportion of documents belong to category c_k in the whole corpus. The other is the Maximum method $f_{max}(t_i) = \max_{k=1}^{c}\{f(t_i, c_k)\}$, which ranks terms according to their maximum weights across all categories. Among these three methods, [6] prefer the salient terms which are unique from one category to another, that is the Maximum approach. [3] also declare that the Maximum method outperformed both the Summation method and the Average method, but the experimental results are not given. Therefore, a question arises here, " Can we perform extensive experimental comparison of these methods, moreover, can we propose a better method than these existing ones?"

In our research, based on existing supervised term weighting methods, we propose a new term ranking method, the Maximum Gap. We illustrate by concrete example that this method can better distinguish those terms which can better differentiate one or more categories from the others than some existing ones, namely, the Summation, Maximum, and Average methods (see [3,6,4]). We conduct a series of comparative experiments on the Reuter-21578 text corpus. SVM and Boostexter are adopted as the learning algorithms. Average precision is used as the evaluation method. In our numerical experiments, Maximum Gap outperforms the other three term ranking methods.

This paper is organized as follows: In Section 2 we survey the existing term-category weighting methods and term ranking methods, then we propose a new term ranking method. In Section 3 we describe the detailed experimental settings. In Section 4 we report experimental results and discussion. We conclude in Section 5.

2 A New Feature Ranking Method

In this section, we review existing term weighting methods (information gain and χ^2 statistic) as well as term ranking methods (Summation, Maximum, and Average methods), introduce a new term ranking method for text categorization, and provide an example to demonstrate the effectiveness of the new method on Reuter-21578 text corpus.

2.1 Term-Category Weighting

Over the years, a number of methods have been developed to measure the discriminating power of various terms with respect to different categories, such as

information gain, odds ratio, and χ^2 statistic. In this research, we discuss information gain and χ^2 statistic, which have been shown to be effective for text categorization [4].

For term t_i and class c_k, the general functions of information gain and χ^2 statistic can be shown as:

$$IG(t_i, c_k) = P(t_i, c_k) \log \frac{P(t_i, c_k)}{P(t_i)P(c_k)} + P(\overline{t}_i, c_k) \log \frac{P(\overline{t}_i, c_k)}{P(\overline{t}_i)P(c_k)} \tag{1}$$

$$\chi^2(t_i, c_k) = \frac{n[P(t_i, c_k)P(\overline{t}_i, \overline{c}_k) - P(t_i, \overline{c}_k)P(\overline{t}_i, c_k)]^2}{P(t_i)P(\overline{t}_i)P(c_k)P(\overline{c}_k)} \tag{2}$$

where $P(t_i, c_k)$ denotes the probability a document is from category c_k when term t_i occurs at least once in it, $P(t_i, \overline{c}_k)$ denotes the probability a document is not from category c_k when term t_i occurs at least once in it, $P(\overline{t}_i, c_k)$ denotes the probability a document is from category c_k when term t_i does not occur in it, $P(\overline{t}_i, \overline{c}_k)$ denotes the probability a document is not from category c_k when term t_i does not occur in it, n denotes the number of documents.

Text categorization problems on multi-class datasets can be simplified into multiple independent binary classification problems. In each experiment, a chosen category c_k can be tagged as 1, and the other categories in the same corpus are combined together as 0. A contingency table (see Table 1) can be used to record the number of documents which contain term t_i and do not contain term t_i under category c_k and $\overline{c}_k$, and the sum of these four elements, n, is the number of documents of the dataset.

Table 1. The contingency table for category c_k and term t_i

	t_i	$\overline{t}_i$
Positive Category: c_k	a	b
Negative Category: $\overline{c}_k$	c	d

Notation:
a: Number of documents in class c_k that contain term t_i
b: Number of documents in class c_k that do not contain term t_i
c: Number of documents in class $\overline{c}_k$ that contain term t_i
d: Number of documents in class $\overline{c}_k$ that does not contain term t_i

[6] use these four elements in Table 1 to estimate the probabilities in formula (1) and (2). The functions of information gain and χ^2 are rewritten as:

$$IG(t_i, c_k) = \frac{a}{n} \log \frac{an}{(a+b)(a+c)} + \frac{c}{n} \log \frac{cn}{(c+b)(a+c)} \tag{3}$$

$$\chi^2(t_i, c_k) = \frac{n(ad - bc)}{(a+c)(b+d)(a+b)(c+d)} \tag{4}$$

2.2 Maximum Gap

The formulas (3) and (4) define weights for each term t_i according to different categories $c_k, k = 1, \cdots, c$. We denote these weights by $f(t_i, c_k)$. In this paper, two cases will be considered: $f_{ig}(t_i, c_k) = IG(t_i, c_k)$ and $f_{\chi^2}(t_i, c_k) = \chi^2(t_i, c_k)$. To rank all the terms, we need to define a weight for each term t_i with respect to all categories. As mentioned before we will investigate three different methods – Maximum, Summation, and Average methods defined by: (see [3,6,4])

$$f_{max}(t_i) = \max_{k=1}^{c}\{f(t_i, c_k)\} \tag{5}$$

$$f_{sum}(t_i) = \sum_{k=1}^{c} f(t_i, c_k) \tag{6}$$

$$f_{avg}(t_i) = \sum_{k=1}^{c} P(c_k) f(t_i, c_k) \tag{7}$$

In this section, we propose a new term ranking method that will be called Maximum Gap (MG). Unlike the above approaches, this method aims to distinguish, in terms of weights, those terms which can better differentiate one or more categories from the others.

First, we organize term i's weights $\{f(t_i, c_k)\}_{k=1}^{c}$ as follows:

$$f(t_i, c_{k_1}) \geq f(t_i, c_{k_2}) \geq \cdots f(t_i, c_{k_c})$$

then the Maximum Gap of term t_i is defined as

$$f_{mg}(t_i) = \max_{j=1}^{c-1}\{f(t_i, c_{kj}) - f(t_i, c_{k_j+1})\} \tag{8}$$

In the following example, we demonstrate why MG might be more efficient than the other three methods.

Example: From Reuter-21578 corpus, we select the top 30 terms selected by different term ranking methods. For both clarity and briefness, we only compare Maximum Gap and Maximum methods, where terms are weighted by information gain. The Maximum method is chosen because it is accepted that (see for example, [3]) this method is better compared to Summation and Average methods. Note that our experimental results in Section 4 also support this opinion.

Among the top 30 selected terms ranked by Maximum Gap and Maximum methods respectively, Table 2 lists 14 terms that are not selected by the opposite term ranking method (those 23 terms selected by both methods are omitted).

It can be seen that the terms exclusively selected by the Maximum Gap method are more closely related to the top 10 categories (see Table 2) than those terms that selected by Maximum method. For example, **us** (this refers to us or USA), **the**, **central**, and **note** selected by Maximum method are less related to the top 10 categories, while all terms selected by Maximum Gap

Table 2. Terms exclusively selected by Maximum Gap and Maximum term weighting methods out of top 30 terms. The top10 categories of Reuter-21578 are acq, corn, crude, earn, grain, interest, money-fx, ship, trade, and wheat.

	Maximum	Maximum Gap
1	us	surplu
2	the	petroleum
3	market	acquisit
4	loss	bui
5	export	tariff
6	central	yen
7	note	energi

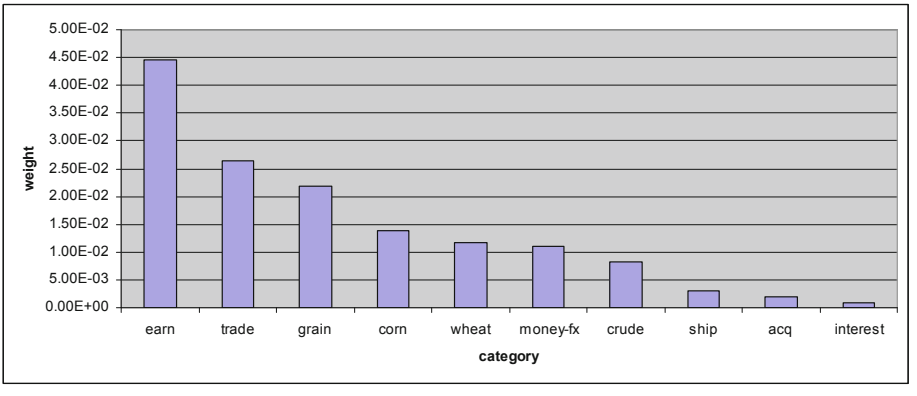


Fig. 1. Term-Category weights (calculated by information gain, Equation 3) of term "us" in Reuter-21578 top 10 categories (ordered by weights)

method are closely related to the top 10 categories. Here we should note that the selected terms have been stemmed by Porter Stemmer [7], for example, the original spelling of **surplu**, **bui**, and **energi** are **surplus**, **buy**, and **energy**. This concrete example can give us a direct explanation why Maximum Gap method might be better than Maximum method.

To provide insight into the performance of Maximum Gap, as an example we investigate the weights $f_{ig}(t_i, c_k)$ of term **us** and term **yen**[1], with respect to the top 10 categories. These two terms are selected respectively by Maximum method and Maximum Gap method. Fig. 1 and 2 lists the sorted weights of **us** and **yen** with respect to top 10 categories. Compared with term **us**, the

[1] In the data set, term **us** appears 6216 times representing the *United State (U.S.)* (5767 times), the *US Dollar* (171 times), and the word *us* (278 times) respectively. Term **yen** appears 629 times, which stands for Japanese currency only. It is clear that **yen** is a more predictable term, which is highly related to the category Money Foreign Exchange (money-fx), while term **us** appears frequently almost in all categories (See Figure 1, and Figure 2).

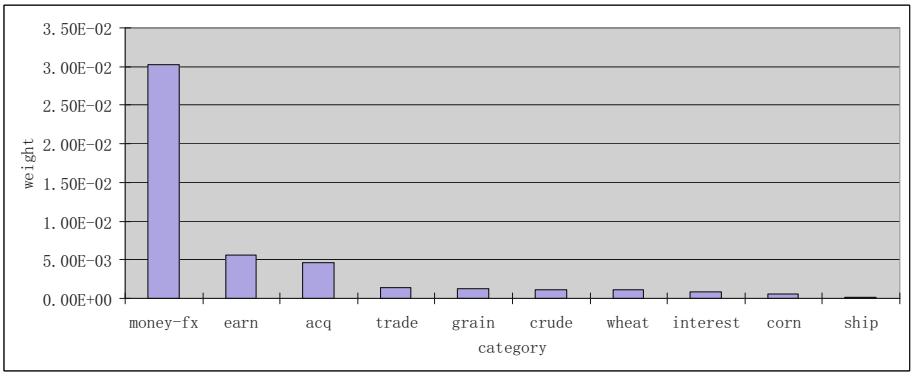


Fig. 2. Term-Category weights (calculated by information gain, Equation 4) of term "yen" in Reuter-21578 top 10 categories (ordered by weights)

Maximum Gap of term **yen** is bigger, but Maximum, Summation, and Average values are smaller. This is why term **us** is selected by Maximum, while term **yen** is selected by Maximum Gap.

3 Experiments

In this section, we describe the relevant details related to our experiments.

Dataset: In this controlled experiment, Reuter-21578 [8] is adopted as the benchmark dataset. In particular, the documents of the top 10 topics are extracted, from which 9393 related documents out of 21578 documents are extracted. Taking into account a large number of documents (9393) in the data set, we use 4-fold cross validation for the purpose of evaluation. Because documents are multilabel, we arrange these folds as follows. The first step, we consider all the combinations of multi-labeled classes and partition them based on the classes they belong to. The second, we fold each of the partitions, rather than the entire dataset, so that we could always keep the pattern for a particular classs combination from the testing set in the training set.

Learning Algorithms: In terms of learning algorithms, SVM and Boostexter are selected. For SVM, we use Chih-Jen Lin's LIBSVM (see [9]). Boostexter (see [10]) is based on the Boosting concept in Machine Learning. It has been proved as one of the most efficient classification algorithms and widely applied in many areas. Both SVM and Boostexter have shown competitive performance on text categorization [11,10].

Supervised Term Weighting Methods: In our experiment, we select two methods to weight the terms across different categories, namely information gain given by Equation (3) and the χ^2 statistic given by Equation (4). These methods have been shown effective and suitable for text categorization [4].

Performance Evaluation: In general, recall, accuracy or confusion matrix are used to evaluate the performance of the classification. These approaches are commonly used for binary or multiclass classification, where correct or not correct results can be evaluated without difficulty. According to multilabel classification problem, the ranking information of the predicted labels are also need to be considered. Average precision [12,10] is an evaluation method that are designed for multilabel classification problems, where the degree of accuracy can be measured by a single number that is more convinient for comparison purposes. Average precision is a performance measure previously used for evaluation of information retrieval (IR) systems [13]. In our experiments, we use a modified Average Precision (see [14]).

Given classifier (h, H) , predicted labels are denoted by $\mathcal{H}(x)$, actual labels are denoted by $\mathcal{Y}(x)$. Let $Y(x) = \{l \in \{1, \ldots, c\} : \mathcal{Y}_l(x) = 1\}$ be the set of actual labels of document x and $\mathcal{H}(x) = \{\mathcal{H}_1(x), \cdots, \mathcal{H}_c(x)\}$ be predicted labels. We denote by $\mathcal{T}(x)$ the set of all ordered labels $\tau = \{i_1, \ldots, i_c\}$ satisfying the condition

$$\mathcal{H}_{i_1}(x) \geq \ldots \geq \mathcal{H}_{i_c}(x);$$

where $i_k \in \{1, \ldots, c\}$ and $i_k \neq i_m$ if $k \neq m$.

In the case, when the numbers $\mathcal{H}_i(x), \quad i = 1, \cdots, c,$ are different, there is just one order τ satisfying this condition. But if there are labels having the same value then we can order the labels in different ways; that is, in this case the set $\mathcal{T}(x)$ contains more than one order.

Given order $\tau = \{\tau_1, \ldots, \tau_c\} \in \mathcal{T}(x),$ we define the rank for each label $l \in Y(x)$ as $rank_\tau(x; l) = k$, where the number k satisfies $\tau_k = l$. Then *Precision* is defined as:

$$P_\tau(x) = \frac{1}{|Y(x)|} \sum_{l \in Y(x)} \frac{|\{k \in Y(x) : \ rank_\tau(x; k) \leq rank_\tau(x; l)\}|}{rank_\tau(x; l)}.$$

Here, we use the notation $|S|$ for the cardinality of the set S. This measure has the following meaning. For instance, if all observed labels $Y(x)$ have occurred on the top of ordering τ then $P_\tau(x) = 1$. Clearly the number $P_\tau(x)$ depends on order τ. We define

$$P_{best}(x) = \max_{\tau \in \mathcal{T}(x)} P_\tau(x) \quad \text{and} \quad P_{worst}(x) = \min_{\tau \in \mathcal{T}(x)} P_\tau(x),$$

which are related to the "best" and "worst" ordering. Therefore, it is sensible to define the *Precision* as the midpoint of these two versions:

$$P(x) = \frac{P_{best}(x) + P_{worst}(x)}{2}.$$

Average Precision over all records $\mathcal{X}$ will be defined as:

$$P_{av} = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} P(x).$$

4 Experimental Results

In this Section, we present the experimental results and compare the performance of the four term ranking methods discussed above.

To compare the performance of different term ranking methods, we use filter methods to select terms and test by SVM and Boostexter. We use S_n to denote the set of the top n terms ranked by certain term ranking methods. In our experiments, if a very small term subset is adopted, many documents of the corpus can not be included in training and test sets. Different term weighting methods can have different training and test subsets included, and they can not be compared appropriately. Actually, in our experiments, the selected terms can cover almost all the documents of our corpus if we have more than 20 terms selected. We only select 9 groups of sequential term subsets from Reuter-21578 corpus $S_{20} \subset S_{30} \subset S_{40} \subset S_{50} \subset S_{60} \subset S_{70} \subset S_{80} \subset S_{90} \subset S_{100}$.

The performance of four term ranking methods is shown in Table 3. Information gain and χ^2 statistic are used to weight terms across all categories respectively. Therefore, we have four different combinations of term weighting methods and text categorization algorithms shown in four columns in Table 3. The value in the table is the average performance among 9 groups of term subsets. The highest value according to different ranking methods is indicated by bold font.

In Table 4, 5, and 6, we make pairwise comparisons of different term ranking methods. The numbers in the second and the third columns of Table 4, 5, and 6 describe how many groups of term subsets show better evaluation performance.

Table 3. Terms are weighted by information gain and χ^2 statistic respectively. SVM and Boostexter are adopted as text categorization algorithms. Text categorization performances are evaluated by average precision. The average performance of 9 groups of term subsets are calculated according to four term ranking methods.

	IG-SVM	χ^2-SVM	IG-Boostexter	χ^2-Boostexter
mg	**93.751**	**93.264**	**95.345**	**95.121**
max	93.601	93.058	95.226	94.782
sum	93.128	92.395	94.923	94.493
avg	92.397	93.131	94.039	95.120

Table 4. Pairwise comparison of Maximum Gap and Maximum on 9 different term sets. 2 different term weighting methods (IG and χ^2 statistics) and 2 categorization algorithms (SVM and Boostexter) applied. In total, Maximum Gap perform better in 25 cases out of 36.

	Maximum Gap	Maximum
IG-SVM	**5**	4
IG-Boostexter	**7**	2
χ^2-SVM	**6**	3
χ^2-Boostexter	**7**	2
total	**25**	11

Table 5. Pairwise comparison of Maximum Gap and Summation on 9 different term sets. 2 different term weighting methods (IG and χ^2 statistics) and 2 categorization algorithms (SVM and Boostexter) applied. In total, Maximum Gap perform better in 34 cases out of 36.

	Maximum Gap	Summation
IG-SVM	**8**	1
IG-Boostexter	**9**	0
χ^2-SVM	**8**	1
χ^2-Boostexter	**9**	0
total	**34**	2

Table 6. Pairwise comparison of Maximum Gap and Average on 9 different term sets. 2 different term weighting methods (IG and χ^2 statistics) and 2 categorization algorithms (SVM and Boostexter) applied. In total, Maximum Gap perform better in 27 cases out of 36.

	Maximum Gap	Average
IG-SVM	**9**	0
IG-Boostexter	**5**	4
χ^2-SVM	**9**	0
χ^2-Boostexter	4	**5**
total	**27**	9

In all of our controlled experiments, Maximum Gap outperforms other term ranking methods in terms of the average performance of the 9 selected feature subsets. In the pairwise comparison with the existing methods, Maximum Gap method also performs very well. The only exception is the comparison with Average method by χ^2-Boostexter (see Table 6), but the difference between them is very close (4 to 5).

5 Conclusion

We present a new term ranking method for text categorization that is called Maximum Gap. This method is compared with three other similar methods: Maximum, Summation, and Average methods. Numerical experiments are carried out on the Reuter-21578 dataset. Experimental results show that the Maximum Gap outperforms other term ranking methods in selecting better terms for the text categorization task.

References

1. Lan, M., Tan, C.L., Low, H.-B.: Proposing a new term weighting scheme for text categorization. In: AAAI. AAAI Press, Menlo Park (2006)
2. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing & Management 24(5), 513–523 (1988)

3. Debole, F., Sebastiani, F.: Supervised term weighting for automated text categorization. In: SAC, pp. 784–788. ACM, New York (2003)
4. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Fisher, D.H. (ed.) ICML, pp. 412–420. Morgan Kaufmann, San Francisco (1997)
5. Duch, W., Duch, G.: Filter methods. In: Feature Extraction, Foundations and Applications, pp. 89–118. Physica Verlag, Springer (2004)
6. Liu, Y., Loh, H.T., Youcef-Toumi, K., Tor, S.B.: Handling of Imbalanced Data in Text Classification: Category-Based Term Weights. In: Kao, A., Poteet, S.R. (eds.) Natural Language Processing and Text Mining, p. 171 (2006)
7. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
8. Lewis, D.D.: Reuters-21578 text categorization test collection. Distribution 1.3 (2004)
9. Hsu, C.W., Chang, C.C., Lin, C.J., et al.: A practical guide to support vector classification (2003)
10. Schapire, R.E., Singer, Y.: Boostexter: A boosting-based system for text categorization. Machine Learning 39(2/3), 135–168 (2000)
11. Joachims, T., Nedellec, C., Rouveirol, C.: Text categorization with support vector machines: learning with many relevant. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
12. Li, T., Zhang, C., Zhu, S.: Empirical studies on multi-label classification. In: ICTAI, pp. 86–92. IEEE Computer Society, Los Alamitos (2006)
13. Salton, G.: Developments in automatic text retrieval. Science 253(5023), 974–980 (1991)
14. Mammadov, M.A., Rubinov, A.M., Yearwood, J.: The study of drug-reaction relationships using global optimization techniques. Optimization Methods and Software 22(1), 99–126 (2007)

A Comparative Study of a Practical Stochastic Clustering Method with Traditional Methods

Swee Chuan Tan[1], Kai Ming Ting[2], and Shyh Wei Teng[2]

[1] SIM University, Clementi, Singapore 599490
jamestansc@unisim.edu.sg
[2] Monash University, Churchill, Vic 3842, Australia
{kaiming.ting,shyh.wei.teng}@monash.edu

Abstract. In many real-world clustering problems, there usually exist little information about the clusters underlying a certain dataset. For example, the number of clusters hidden in many datasets is usually not known *a priori*. This is an issue because many traditional clustering methods require such information as input. This paper examines a practical stochastic clustering method (PSCM) that has the ability to find clusters in datasets without requiring users to specify the centroids or the number of clusters. By comparing with traditional methods (k-means, self-organising map and hierarchical clustering methods), the performance of PSCM is found to be robust against overlapping clusters and clusters with uneven sizes. The proposed method also scales well with datasets having varying number of clusters and dimensions. Finally, our experimental results on real-world data confirm that the proposed method performs competitively against the traditional clustering methods in terms of clustering accuracy and efficiency.

Keywords: Data Clustering, Stochastic Clustering.

1 Introduction

Cluster analysis involves grouping data objects into clusters, having similar objects within a cluster that maximises inter-cluster differences [4]. Many approaches have been proposed over the years. Traditional clustering methods, such as K-Means [8], Self-Organizing Map (SOM) [6], and Hierarchical clustering methods [5], are well-known. One key problem with many traditional clustering methods is that they require the user to input information about the cluster structure before a dataset can be analysed. This requirement is not practical because such information may not be available prior to cluster analysis. For example, K-Means requires users to specify the number of clusters as an input to the algorithm; but the number of clusters is often not known *a priori*.

This paper examines a practical stochastic clustering method (PSCM), a practical implementation of a General Stochastic Clustering Framework that has already been shown to outperform several Swarm-based Clustering methods [12]. This paper reports a study on the relative performance of PSCM with respect to three popular traditional clustering methods.

J. Li (Ed.): AI 2010, LNAI 6464, pp. 112–121, 2010.

Unlike K-Means, PSCM does not require users to specify the number of clusters before the clustering process. This method differs from SOMs [6] because it does not use a map (or grid) in its formalism. Hence PSCM does not require users to define parameters associated with the gird (e.g., grid size and the neighbourhood function). PSCM is different from Hierarchical clustering because it does not perform a deterministic search of similar clusters using predefined linkage metric. Instead, data items are moved from one cluster to another using a stochastic heuristic. As such, PSCM does not concern itself with measuring inter-cluster similarity using linkage metric; thus it does not need to choose the correct linkage metric in different situations.

The structure of this paper is as follows. We first review the traditional methods, and then we describe the PSCM algorithm. In our experiments, we show that PSCM performs better than three traditional clustering methods. Finally, we provide some discussions and conclude this paper.

2 Traditional Clustering Methods

This section gives a quick review of three traditional clustering methods; more details of these methods can be found in many texts (e.g., [11]).

K-Means Clustering. K-Means [8] is an iterative algorithm that divides a dataset into a pre-specified number of clusters (K). In each iteration, K-Means assigns each item to its nearest cluster centre (or centroid) so as to minimise the distance between each item and its centroid. Once all the items have been assigned, the new centroids are computed again and the next iteration repeats the same process of assigning the items to their nearest centroids. K-Means stops when the centroids stop changing.

K-Means is simple and has been used in various applications. For example, in Bioinformatics, K-Means was used to identify transcriptional sub-networks in yeast [13].

Hierarchical Clustering. The aim of hierarchical clustering is to organise a set of data as a hierarchical tree; objects with similar characteristics are kept in the same branch of the tree, while objects with different characteristics are placed in different branches. There are two directions in which hierarchical clustering can be performed: (i) bottom-up: where data items are gradually merged (or agglomerated); and (ii) top-down: where data groups are gradually divided into smaller sub-groups.

Agglomerative methods generally produce good results. The hierarchical tree provides a natural view of the clusters and allows the user to explore the structure of data at different levels of granularities. One example is exploring clusters in vast biological datasets [9].

Self-Organizing Map Clustering. Through the use of self-organizing artificial neural networks [6], SOMs create a map of one or two dimensions, displaying

clusters of high-dimensional data. Similar to K-Means, users must provide SOM with the number of clusters in a dataset. Once this information is provided, the initial clusters are filled with data randomly selected from the data space; and then the homogeneity of each cluster is improved by grouping similar items close to one another on the map. Hence, SOM is also a tool for visualizing complex data in high dimensional space. A very good example application of SOM is gene expression data analysis [10].

As mentioned, most traditional clustering methods require information pertaining to cluster structure which is usually not known before the clustering process. In the next section, we present PSCM, which is set out to address this issue.

3 Practical Stochastic Clustering Method

Figure 1 illustrates the clustering process of the proposed Practical Stochastic Clustering Method (PSCM). As an example, this figure shows a simple task of clustering a dataset D with n items, where each item *belongs to a known type*. PSCM begins by allocating each item to one bin. Then, the main loop of PSCM is repeated for a fixed number of iterations. In each loop, two different items i and j are randomly selected from D. If i and j are of the same type, then it is necessary to move an item from one bin to another. Since each loop involves searching for two similar items and grouping them together, groups of similar items (i.e., clusters) emerge as the procedure is repeated many times. A more formal description of this process is presented in Algorithm 1.

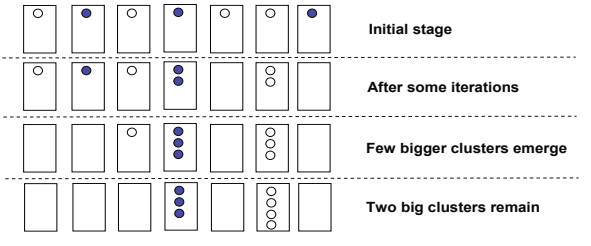


Fig. 1. An illustration of the clustering process

The direction of movement from one bin to another, is based on a concept called *level of support*. An item with a low level of support will move to the bin of another item that has a higher level of support. Formally, the level of support for an item x is defined as:

Definition 1: *Level of Support*

Let S_x be a set of all items similar to item x in D, excluding x. Let X be the set of items within x's bin. Then, the *Level of Support* is defined as $c(x) = |A_x|/|S_x|$, where $A_x = S_x \cap X$.

Algorithm 1. Practical Stochastic Clustering Method

Input: dataset of n items, D
Output: a set of clusters
Estimate the similarity thresholds for n items
Initialise n bins by allocating each item in D to a bin
for $iteration = 1$ to $maxIteration$ **do**
  $i :=$ randomSelect(D)
  $j :=$ randomSelect(D), where $i \neq j$
  if $acceptance(i, j)$ **then**
    Store the comparison outcome in V_i and V_j
    $(\hat{c}(i) < \hat{c}(j))$? $move(i, j) : move(j, i)$
  end if
end for
Return all non-empty bins as a set of final clusters

Literally, $|A_x|$ is the number of items similar to x *within its current bin*, excluding x. Let $B_x = S_x \backslash A_x$, then $|B_x|$ is the number of items similar to x *outside of its current bin*. If $|A_x| \gg |B_x|$, then $c(x)$ is close to one, indicating that x has a high level of support in its current bin and x is likely to attract similar items from other bins. On the other hand, if $|A_x| \ll |B_x|$, then $c(x)$ is close to zero, indicating that x has a low level of support and x is likely to move to other bins which contain similar items. When two bins have the same level of support, we move an item from one bin to another stochastically.

Estimating the Level of Support. During the PSCM clustering process, there is a series of comparisons. Each comparison involves two separate items that are randomly selected. The idea is to store the outcomes of the last b *recorded* comparisons made with each item, and estimate $c(\cdot)$ based on these outcomes.

The outcomes can be recorded using a first-in-first-out register, denoted as V_x, which has a fixed size of b. Each element of V_x stores one bit: (i) when y is in the same bin as x, a '1' is recorded; or (ii) when y is not in the same bin as x, a '0' is recorded. Finally, $c(x)$ at current time t is estimated as $\hat{c}_t(x) = \frac{1}{b} \sum_{k=0}^{b-1} V_x[t-k]$, where $V_x[t-k]$ is the comparison outcome recorded at k time-units before the current time t. Hence, $\hat{c}_t(x)$ is the proportion of ones in V_x at time t. The sample size b is set at 30, which is sufficiently large for a simple random sample.

Clustering real-world Data. To handle numerical data, we use the *acceptance* predicate proposed by some researchers (e.g., [7]), which is defined as follows:

$$acceptance(i, j) = sim(i, j) > max(T_i, T_j), \tag{1}$$

where $sim(i, j) = 1 - d(i, j)$; and $d(i, j)$ is a normalised 1-norm distance metric between i and j, scaled to [0, 1]. Thus $sim(i, j) = 0$ if i and j are completely dissimilar, and $sim(i, j) = 1$ if i and j are the same. In general, $acceptance(i, j)$ is true only if the similarity $sim(i, j)$ is greater than the higher similarity threshold $max(T_i, T_j)$.

The similarity threshold for item x (denoted as T_x) is commonly used in Swarm-based Clustering (SBC) methods (e.g., [2]); but we generalise it as:

$$T_x = \omega \cdot meanSim(x, \cdot) + (1 - \omega) \cdot maxSim(x, \cdot), \tag{2}$$

where ω is a weight in the range of [0, 1]. We set ω at 0.3 for all the 25 datasets used in our experiments. $meanSim(x, \cdot)$ is the mean similarity between item x and all the other items in a dataset; and $maxSim(x, \cdot)$ is the maximum similarity between item x and all the other items in the dataset.

We can estimate the similarity threshold of each data item before the clustering process [7]. Each estimation involves comparing an item with 150 other items randomly selected from the dataset.

The maximum number of iterations is defined as $k \cdot n$, where k is fixed at 2000 for all the experiments reported in this paper. After the maximum number of iterations has been reached, each item in the smaller bins (with bin size less than a threshold) is reassigned to the bin that consists of items with a centroid that is most similar to it. In PSCM, the bin size threshold is $min(50, \frac{n}{20})$; the threshold of $\frac{n}{20}$ is based on the criterion used by Fabien et al. [2]. When n is large, we limit the bin size threshold to 50.

4 Experiments and Results

This section describes the experimental setup and then compares PSCM with the three traditional clustering methods, K-Means, AvgLink and SOM.

Experimental setup. To test the robustness of the algorithms, we use four test scenarios previously adopted by Handl [3]: (i) Squares series—datasets with increasingly overlapping clusters; (ii) Sizes Series—datasets with increasingly unevenly-sized clusters; (iii) datasets with different number of clusters, sizes, shapes and dimensions; and (iv) seven real-world datasets taken from the UCI Machine Learning Repository [1].

We use the results of K-Means, AvgLink and SOM that were previously generated by Handl [3]. In their experiments, K-Means and SOM were given a 'privilege' of knowing the actual number of clusters, while AvgLink was given the appropriate stopping criterion so that it would stop the agglomeration process when the correct number of clusters was reached. Furthermore, Handl et al. conducted 1000 runs of K-Means on each dataset, and chose only the best clustering result in every 20 runs; this produced 50 best results for each dataset.

To be consistent with the experimental settings used by Handl, we conducted 50 independent runs of PSCM on each dataset. The 50 results for each dataset were obtained and finally evaluated using the average runtime taken (in seconds) and the average *F-measure* [14]. The *F-Measure* quantifies how well a set of generated clusters match the actual classes in the dataset. It is in the interval [0, 1] and equals 1 for perfect clustering.

Test Scenario 1: Increasingly Overlapping Clusters We use seven synthetic datasets known as Square series, four of which are shown at the top of Figure 2.

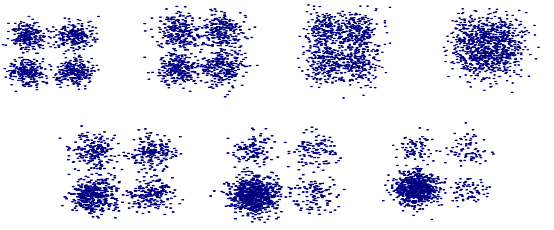
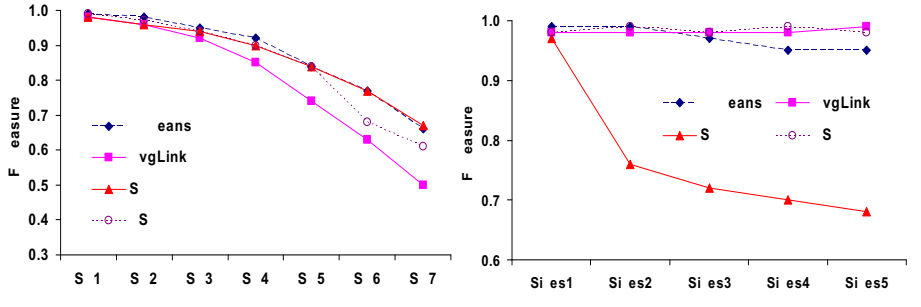


Fig. 2. Each dataset contains 1000 instances and four bivariate Gaussian clusters. (Top) Square1, Square3, Square5 and Square7 contain increasingly overlapping clusters. (Bottom) Sizes1, Sizes3, Sizes5 contains clusters with increasing uneven sizes.

Figure 3(a) shows that initially all algorithms have high F-Measures because the clusters are well separated in the Square1 dataset. As the clusters become more overlapped, the performance of the algorithms begin to degrade. AvgLink gives the lowest F-Measure from Square1 to Square7. PSCM is among the top two methods, with its performance similar to that of K-Means and SOM from Square1 to Square5. SOM performs well due to its topology preserving property [3]. K-Means is among the top few models because, as mentioned earlier, it always selects the 50 best results from 1000 runs on each dataset.



(a) Increasingly overlapping clusters (b) Increasing uneven cluster sizes

Fig. 3. Evaluating the algorithms using F-Measure. Square is abbreviated as 'Sq' in (a).

Test Scenario 2: Increasingly Uneven Cluster Sizes Here we use five synthetic datasets known as the Sizes series; three of which are shown at the bottom of Figure 2.

Figure 3(b) shows that, initially all algorithms have relatively high F-Measures when the dataset is Sizes1. However, as the cluster sizes become more uneven, the performance of the algorithms, evaluated by F-Measure, degrades. SOM consistently gives the lowest F-Measure. The next worst performing method is K-Means because this method is known to be sensitive to clusters with different sizes. When Sizes4 and Sizes5 are used, its F-Measure reduces because it sometimes fails to find the centroids in the smaller clusters. Finally, the performance of AvgLink and PSCM are comparable in terms of F-Measure.

Test Scenario 3: Runtime Scalability Study. We use a series of datasets denoted as xDyC. Here, x denotes the number of dimensions (D), which is in {2, 10, 100}; and y denotes the number of Gaussian clusters (C), which is either 4 or 10. This forms six possible combinations of dimensionality and cluster number in the xDyC datasets. For each combination (e.g., 2D4C), 50 datasets of different cluster shapes and sizes are generated randomly [3]. Hence the clustering algorithms are expected to deal with 300 datasets with different number of clusters, dimensionality, sizes and shapes. In general, datasets with ten clusters will have a lot more data than those datasets with four clusters.

Figure 4 shows that the runtime of K-Means increases as the dimensionality and the number of clusters increase. Its runtime increases drastically on the 100D10C datasets because it needs to recompute high dimensional centroids during each iteration. As for AvgLink, its runtime on the larger datasets (with 10 clusters) is a lot higher than the smaller datasets (with only 4 clusters) due to its quadratic runtime complexity. Notice that SOM also exhibits similar runtime behaviour compared to agglomerative clustering, though to a lesser extent.

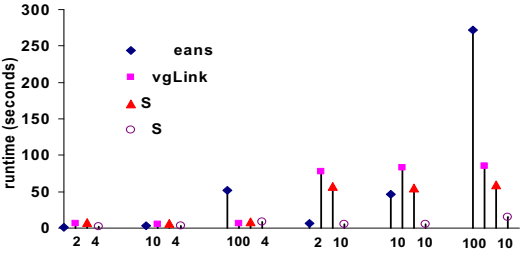


Fig. 4. Average runtime performance of different clustering methods under the conditions of different number of dimensions and clusters in the xDyC datasets

Figure 4 shows that PSCM[1] is the fastest method in most cases. Unlike K-Means, PSCM does not need to recompute the centroids of each cluster, and its runtime is not sensitive to the number of clusters. Figure 4 also shows that PSCM is faster than AvgLink and SOM on larger datasets with ten clusters.

[1] The runtime of PSCM is adjusted based on Handl's computing platform.

Test Scenario 4: Performance on seven real-world datasets The datasets are taken from the UCI Machine Learning repository [1], which include Wisconsin, Dermatology, Zoo, Wine, Iris, Digits and Yeast. A summary of these datasets is given in Table 1.

Table 1. Real-world data. N is the number of instances; N_i is the number of instances for cluster i; D is the number of features; and C is actual number of clusters

Dataset	N	N_i	D	C
Wisconsin	699	458, 241	9	2
Dermatology	366	112, 61, 72, 49, 52, 20	34	6
Zoo	101	41, 20, 5, 13, 4, 8, 10	16	7
Wine	178	59, 71, 48	13	3
Iris	150	50, 50, 50	4	3
Digits	3498	363, 364, 364, 336, 364, 335, 336, 364, 336, 336	16	10
Yeast	1484	463, 429, 244, 163, 51, 44, 37, 30, 20, 5	8	10

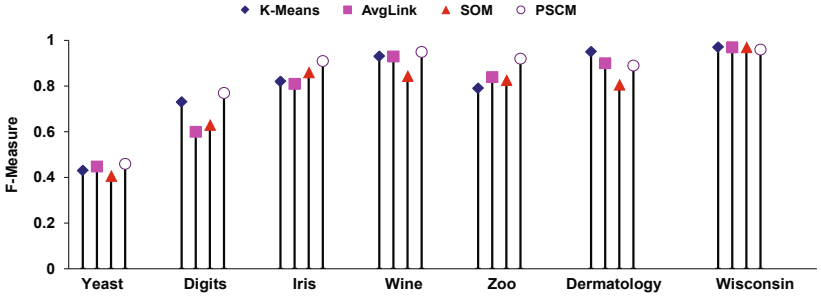


Fig. 5. Average F-Measure of four clustering methods on seven real-world datasets

Figure 5 presents the average F-Measure results of PSCM, K-Means, AvgLink and SOM. Similar to the observation made from results on synthetic data, the traditional methods are unable to perform consistently well throughout all the datasets. For example, K-Means produces the highest and lowest F-Measure on Dermatology and Zoo respectively. Similarly, SOM performs well on the Iris dataset; but gives the worst result on the Dermatology dataset. AvgLink also pales in comparison—its results are the lowest in three of the seven datasets.

Out of the seven real-world datasets, PSCM produces the highest F-Measure on five of them; these datasets include Yeast, Digits, Iris, Wine and Zoo. As for the two remaining datasets, PSCM scores marginally below its rivals.

Discussions. In terms of clustering accuracy, two of the top performers in the first test scenario, SOM and K-Means, were ranked at the bottom in the second test scenario. As for AvgLink, it was the worst performing method in the first

test scenario, and then it rose to the top in the second test scenario. Note that PSCM is the only method that has performed consistently well in both scenarios.

Unlike K-Means, PSCM does not compute cluster centroids. Instead, data bins that contain data close to the centroid are likely to grow into full-fledged clusters. This is because data points around the centroid of a Gaussian cluster will have more similar data in its surroundings compared to data points near the fringe of the cluster. This is so even when the clusters are highly overlapped or when the cluster sizes are highly uneven. This is why PSCM has performed well in both of the above scenarios.

In a separate experiment, we find that varying the values of different parameter settings (ω, k and b) does not affect PSCM's performance drastically. For example, when increasing the maximum iterations $k \cdot n$, we vary k from 1000 to 10000 in steps of 1000; the average F-Measure of PSCM on all datasets only change slightly, from 0.88 to 0.89. This suggests that the PSCM converges after some point and its clustering output remains constant with further iterations.

PSCM can also be used to predict the cluster-membership of a new unseen data point u. A simple way is to assign u to the cluster that has the centroid most similar to u.

5 Concluding Remarks

We have examined a practical stochastic clustering method (PSCM), which has the ability to automatically find the number of clusters in datasets.

In the experiments, three competing traditional clustering methods (K-Means, AvgLink and SOM) were given a 'privilege' of knowing the cluster structure before the cluster analysis process. PSCM was the only method that had to derive the number of clusters by itself. Yet, when tested using 25 datasets, the traditional clustering methods tend to produce mixed results—they can excel or fail under different conditions. In contrast, the performance of PSCM, in terms of clustering accuracy and runtime, is consistently good across most of the datasets.

While the method reported in this paper employs the normalised 1-norm distance metric, our latest investigation suggests that we can remove distance normalisation by reformulating Equations 1 and 2 using dissimilarity function.

Acknowledgment

The first author was partly supported by Monash Postgraduate Publication Award when this research was carried out.

References

1. Blake, C., Merz, C.: UCI repository of machine learning databases. Technical report, Department of Information and Computer Sciences, University of California, Irvine (1998)

2. Fabien, P., Hanene, A., Gilles, V., Christiane, G.: A New Approach of Data Clustering Using a Flock of Agents. Evolutionary Computation 15(3), 345–367 (2007)
3. Handl, J.: Ant-based methods for tasks of clustering and topographic mapping: extensions, analysis and comparison with alternative methods. Master's thesis, University of Erlangen-Nuremberg, Germany (2003)
4. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31(3), 263–323 (1999)
5. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: an Introduction to Cluster Analysis. John Wiley and Sons, Chichester (1990)
6. Kohonen, T.: Self-Organization and Associative Memory. Springer, Berlin (1984)
7. Labroche, N., Guinot, C., Venturini, G.: Fast Unsupervised Clustering with Artificial Ants. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 1143–1152. Springer, Heidelberg (2004)
8. MacQueen, L.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press, Berkeley (1967)
9. Seo, J., Shneiderman, B.: Interactively Exploring Hierarchical Clustering Results. Computer 35(7), 80–86 (2002)
10. Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E.S., Golub, T.R.: Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. Proceedings of the National Academy of Sciences of the United States of America 96(6), 2907–2912 (1999)
11. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley Longman Publishing Co., Inc., Boston (2005)
12. Tan, S.C.: Approaches to simplify and improve swarm-based clustering. PhD's thesis, Monash University, Australia (2009)
13. Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., Church, G.M.: Systematic determination of genetic network architecture. Nature Genetics 22, 281–285 (1999)
14. van Rijsbergen, C.: Information Retrieval, 2nd edn. Butterworths, London (1979)

An Effective Pattern Based Outlier Detection Approach for Mixed Attribute Data

Ke Zhang[1] and Huidong Jin[1,2,⋆]

[1] College of Engineering & Computer Science, Australian National University, Acton ACT 2601, Australia
[2] CSIRO Mathematics, Informatics and Statistics, GPO Box 664, Canberra ACT 2601, Australia
ke.zhang@cecs.anu.edu.au, Warren.Jin@csiro.au

Abstract. Detecting outliers in mixed attribute datasets is one of major challenges in real world applications. Existing outlier detection methods lack effectiveness for mixed attribute datasets mainly due to their inability of considering interactions among different types of, e.g., numerical and categorical attributes. To address this issue in mixed attribute datasets, we propose a novel Pattern based Outlier Detection approach (POD). Pattern in this paper is defined to describe majority of data as well as capture interactions among different types of attributes. In POD, the more does an object deviate from these patterns, the higher is its outlier factor. We use logistic regression to learn patterns and then formulate the outlier factor in mixed attribute datasets. A series of experimental results illustrate that POD performs statistically significantly better than several classic outlier detection methods.

Keywords: outlier detection; mixed attribute data; pattern based outlier detection.

1 Introduction

Of all the data mining techniques that are in vogue, outlier detection comes closest to the metaphor of mining for nuggets of information in real world data. It is concerned with discovering the exceptional behavior of certain objects. Outlier detection techniques have widely been applied in medicine, finance, information security and so on [1,2,3]. In the recent decades, various outlier detection approaches have been proposed, which can be broadly classified into several categories: distribution-based [4], depth-based [5], distance-based (e.g., k-nearest neighbour (KNN) [6] and [7]), clustering-based [8] and density-based (e.g., local outlier factor (LOF) [1]) methods. Most of these methods are designed for homogeneous datasets, i.e., they only contain a single type of attributes like numerical or categorical attributes.

Real world datasets are usually more complex. They often consist of different types of attributes, e.g. categorical (nominal) and numerical (continuous)

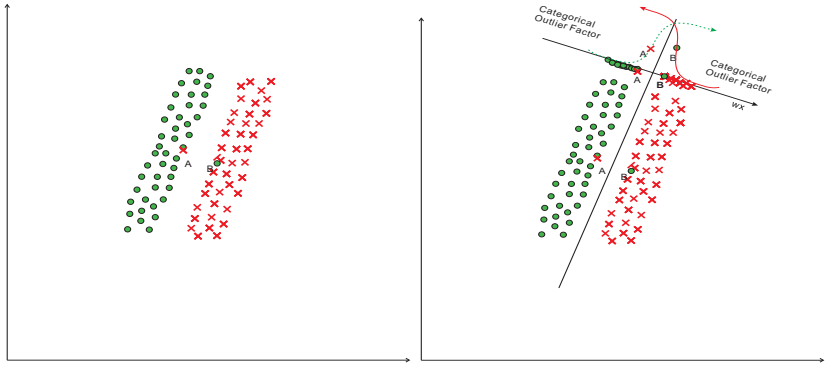
⋆ Corresponding author.

J. Li (Ed.): AI 2010, LNAI 6464, pp. 122–131, 2010.

types, called mixed attribute data. For example, in personal income survey data contain numerical attributes (e.g. working hours per week) and categorical attributes (e.g. occupation). In order to handle mixed attribute data, usually, researchers convert different types of attribute values into a single type. For example, for algorithms working only on numerical attributes, the categorical attribute values have to be recoded as, e.g., 0, 1, 2, $\cdots$, and mixed them with the numerical attribute values of the original dataset [9]. Meanwhile, for some algorithms designed for categorical datasets, numerical values must be discretised into several bins, and treat them as a set of categorical values. However, the recoding and discretising methods for dealing with mixed attribute data have their inherent drawbacks. Firstly, recoding or discretising could introduce noise or lose information in the attribute conversion process [10]. Secondly, improper recoding or discretising settings would significantly deteriorate detection performance. Thirdly, single type attribute outlier detection methods were not often designed for considering interactions between categorical and numerical attributes. Because outlierness or anomaly in mixed attribute datasets is usually resulted from the interaction between categorical and numerical attributes, existing outlier detection methods lack effectiveness on mixed attribute datasets.

In recent years, researchers have proposed several algorithms for dealing with mixed attribute datasets. A typical method, LOADED [10], uses Association Rules to explore infrequent items among categorical values and calculates covariance matrix to examine the anomaly in numerical values. Outliers in mixed attribute datasets are determined by their anomaly scores, which are the sum of anomaly scores in the categorical and the numerical values. Although LOADED gives a specific method for exploring anomalies in either categorical or numerical values, it could not perform quite well due to lack of considering interactions between different types of attributes. Its improved version RELOADED [11] suffers the same problem, though it requires less main memory. More recently, a graph-based outlier detection algorithm [12] has been proposed to separately compute Euclidean distance for numerical values and Hamming distance for categorical values to calculate outlier indicators. A projection-based outlier detection method [13] has been developed which uses the equi-width method to discretise numerical attribute values in order to handle mixed attribute datasets. Although the two recent works claim that they were designed for mixed attribute data, they again consider outlier factors of different types of attributes separately and then sum up them.

In this paper, we propose a Pattern based Outlier Detection approach (POD), which is able to effectively consider interactions between different types of attributes without attribute conversion processes (discretising or recoding). Pattern in this paper is defined to describe majority of data as well as capture interactions among different types of attributes. Then, based on the notation of pattern, a new outlier factor for mixed attribute data is proposed. That is, the more does an object deviate from these patterns, the higher is its outlier factor. In POD, we use logistic regression to acquire patterns and then formulate the outlier factor in mixed attribute datasets. To validate our approach,



(a) A mixed attribute dataset (b) Categorical outlier factor

Fig. 1. Indicative example of pattern, outliers, and COF in a simple dataset

we compare POD with three other typical methods, LOADED [10], KNN [14] and LOF [15] over a series of synthetic and real-world mixed-attribute datasets. Experimental results show statistically significant improvement of POD over the three methods.

The rest of the paper is organised as follows. In Section 2, we introduce patterns and outliers in mixed attribute data and define our outlier factors. In Section 3, the top-n pattern based outlier detection algorithm is presented. Experimental results are reported and compared in Section 4, followed by conclusions in Section 5.

2 Patterns and Outliers in Mixed Attribute Data

Outlierness or anomaly in mixed attribute data are often resulted from interactions between categorical and numerical values. For example, in an income survey dataset, it is common to see that a man has an occupation of engineer and held a Bachelor degree. However, the record becomes unusual if the man is only 10 years old. That is to say, the outlierness or anomaly in mixed attribute datasets has its own characteristics, and it is hard to follow the outlier definition given by single type attribute outlier detection methods, e.g. KNN [6], LOF [15] or existing ones for mixed attribute datasets such as LOADED [10] or a graph based technique [12].

Before exploring a suitable outlier definition, we have to define the normal behaviour or majority in mixed attribute datasets first. We call the normal behaviour or majority in mixed attribute dataset *pattern*. It slightly differs from the ones widely used in pattern recognition community where patterns are regarded as an example, cluster, etc. We give an example to illustrate what a *pattern* looks like in a mixed attribute dataset. Figure 1(a) illustrates a simple mixed attribute dataset, with two numerical attributes and one categorical attribute (binary value). "Dot" objects indicate the data objects with categorical

value "Male", while "cross" for "Female". We can observe that most of "dot" and "cross" objects are regularly located into two groups. We can regard such a data object distribution as a *pattern* in the example, which indicates the normal behaviour or majority in the simple dataset. Intuitively, if an object "looks" like that it does not comply with the pattern, it is suspicious to be an outlier. As shown in Figure 1(a), objects A and B are outliers as they deviate from the pattern. In this work, we concentrate on a specific type of patterns, where only one categorical attribute is involved in one pattern.

We denote $\mathcal{D}$ as a set of mixed attribute data objects. $\mathbf{O}_i \in \mathcal{D}$ is the ith data object. Each data object contains M numerical attributes and N categorical attributes. Denote object i as $\mathbf{O}_i = [\mathbf{x}_i, \mathbf{c}_i]$, where $\mathbf{x}_i = [x_i^1, x_i^2, \cdots, x_i^j, \cdots, x_i^M]$ and $\mathbf{c}_i = [c_i^1, c_i^2, \cdots, c_i^k, \cdots, c_i^N]$, with x_i^j for the jth numerical attribute value and c_i^k for the kth categorical attribute value. To simplify the discussion below, we define $\mathcal{S}'$ a subspace of $\mathcal{D}$ which only contains a subset of attributes in $\mathcal{D}$. $\mathcal{S}'^k$ is a subspace of $\mathcal{D}$ which only contains the kth categorical attribute and all the numerical attributes. $\mathbf{O}_i'^k$ is the projection of $\mathbf{O}_i$ on the subspace $\mathcal{S}'^k$.

Given a mixed attribute subspace $\mathcal{S}'^k$, most of $\mathbf{O}'^k$ exhibit some common characteristics or behaviour. The pattern in mixed attribute subspace is defined as follows:

Definition 1. *We call the common characteristics or behaviour demonstrated by projected objects* $\mathbf{O}'^k$ *as the mixed attribute subspace pattern* P^k.

Based on the pattern definition above, if a mixed attribute data contains N categorical attributes, there are N patterns, $P = \{P^1, P^2, \cdots, P^N\}$. Definition 1 gives us a description of what are the normal objects in mixed attribute space. Furthermore, this definition simplifies the data space a bit. Our pattern definition only focuses on a subspace $\mathbf{S}'$ at a time which only contains one categorical attribute rather than considering the total N categorical attributes in the original data space. Such a simplification provides us a simple mechanism to handle datasets with a large number of categorical attributes. We will use this mechanism to handle a number of categorical attributes in Section 4.

In order to take interactions between categorical and numerical attributes into account, we propose to use logistic regression to acquire patterns in mixed attribute datasets. To simplify our discussion further, we assume all the categorical attributes only have binary value, i.e. $c_i^k \in \{0, 1\}$, in this section. We will handle the general case in Section 3. Given a projected object $\mathbf{O}_i'^k = [\mathbf{x}_i, c_i^k]$ on subspace $\mathcal{S}'^k$ and a binary variable Y where $Y = c_i^k$, a logistic regression might take a simple form like:

$$P_{\mathbf{O}_i'^k} = \begin{cases} P(Y=1|\mathbf{x}_i) = \frac{1}{1+\exp(\mathbf{w}\mathbf{x}_i^T)}, & \text{if } c_i^k = 1, \\ P(Y=0|\mathbf{x}_i) = \frac{\exp(\mathbf{w}\mathbf{x}_i^T)}{1+\exp(\mathbf{w}\mathbf{x}_i^T)} & \text{otherwise,} \end{cases} \tag{1}$$

where $P_{\mathbf{O}_i'^k}$ measures the degree of projected object $\mathbf{O}_i'^k$ complying with pattern P^k, Y_i^k takes the value of c_i^k and $\mathbf{w} = [w_0, w_1, \cdots, w_M]$ is the parameter vector in the logistic regression. $\mathbf{w}$ can be captured from data directly by, e.g., maximising the likelihood [5].

$$\mathbf{w}^k \longleftarrow \arg\max_{\mathbf{w}^k} \prod_i P(c_i^k|\mathbf{x}_i^k, \mathbf{w}^k), \tag{2}$$

where $\mathbf{w}^k$ is the logistic regression parameter in subspace $\mathcal{S}'^k$. As the information of both categorical and numerical attributes is considered in the learning procedure, the pattern can capture some interaction between categorical and numerical attributes.

Based on such a kind of patterns, we can define outliers. More formally, we denote Y is binary variable, governed by a Bernoulli distribution, with parameter $\pi = P(Y = 1)$. x_i^j is the jth element in vector $\mathbf{x}_i$.

Categorical outlier factor. Given a subspace in mixed attribute space, we define *categorical outlier factor* (COF) β to indicate the degree of projected object deviating from its pattern. β_i^k denotes the degree of $\mathbf{O}_i'^k$ deviating from the pattern P^k. It takes the form:

$$\beta_i^k = \begin{cases} \frac{1}{1+\frac{P(Y=1|\mathbf{x}_i)}{1+P(Y=0|\mathbf{x}_i)}} & \text{if } c_i^k = 1, \\ \frac{1}{1+\frac{P(Y=0|\mathbf{x}_i)}{1+P(Y=1|\mathbf{x}_i)}} & \text{otherwise.} \end{cases} = \begin{cases} 1 - \frac{P(Y=1|\mathbf{x}_i)}{2} & \text{if } c_i^k = 1, \\ \frac{1}{2} + \frac{P(Y=1|\mathbf{x}_i)}{2} & \text{otherwise.} \end{cases} \tag{3}$$

In Equation 3, the probability $P(Y = 1|\mathbf{x}_i)$ is used to represent the degree of $\mathbf{O}_i'^k$ deviating from its pattern. For example, if a projected object $\mathbf{O}_i'^k$ with $c_i^k = 1$ is located within the group of other projected objects which have $c_j^k = 0$ ($j \in \{1, \cdots, N\}, j \neq i$), $P(Y = 1|\mathbf{x}_i)$ will be a small value (e.g., very close to 0) and its COF β_i^k will be assigned a large value (e.g., very close to 1). That means this projected object is very likely to contribute to the outlierness of the whole object.

Based on our logistic regression assumption, we can rewrite the expression of β_i^k as follows:

$$\beta_i^k = \begin{cases} 1 - \frac{\exp(\mathbf{w}^k \mathbf{x}_i{}^\top)}{2(1+\exp(\mathbf{w}^k \mathbf{x}_i{}^\top))} & \text{if } c_i^k = 1, \\ \frac{1}{2} + \frac{1}{2(\exp(\mathbf{w}^k \mathbf{x}_i{}^\top))} & \text{otherwise.} \end{cases} \tag{4}$$

The parameter $\mathbf{w}^k$ takes interaction between numerical and categorical information in the learning procedure (see Equation 2). The inner product $\mathbf{w}^k \mathbf{x}_i{}^\top$ further projects the high dimensional data object $\mathbf{O}_i'^k$ onto one dimension space. As shown in Figure 1(b), all mixed attribute data objects in the simple data example are mapped onto the line (indicated by wx) pointing eastern. They are then mapped onto an appropriate COF curve and assigned an outlierness value. By Equation 4, the outlierness of the object which deviates from its pattern will be substantially magnified and can be easily identified. In Figure 1(b), e.g., outliers A and B can be easily highlighted by their reasonably large COF β_i as other objects have relatively small COF.

Mixed attribute data outlier factor. The *categorical outlier factor* (COF) represents some interaction between numerical attributes and one categorical attribute, following a logistic regression setting. In order to consider the interactions with all the categorical attributes, we further introduce *mixed attribute*

data outlier factor (MADOF) which represents the outlierness of the mixed attribute object $\mathbf{O}_i$. We construct a vector $\zeta_i = [\gamma_i, \beta_i^1, \beta_i^2, \cdots, \beta_i^N]$, consisting of all the COF β_i^k $(k = 1, \cdots, N)$ and $\gamma_i = \exp(1 + KDist_i)$ where $KDist_i$ is the kth-nearest neighbour distance [14] of the $\mathbf{O}_i$. γ_i can represent its outlierness in the pure numerical attribute space. Then, MADOF is defined as

$$\text{MADOF}_i = \|\zeta_i\|_2 = \sqrt{\sum_{j=1}^{N+1} (\zeta_i^j)^2}, \tag{5}$$

In the outlierness space $\mathcal{A}$ spanned by ζ, MADOF_i can be regarded as the distance from ζ_i for Object $\mathbf{O}_i$ to the origin of $\mathcal{A}$. The further is ζ_i away from the origin of the outlierness space, the more likely the object to be an outlier in the mixed attribute space. Based on this proposed MADOF for mixed attribute datasets, we will give a Pattern based Outlier Detection (POD) algorithm in the following section.

3 Pattern Based Outlier Detection Algorithm

Map into binary attribute space. In the previous section, our discussion is mainly based on binary categorical values to facilitate formulation. In order to generalise our algorithm, we firstly introduce a mapping mechanism which would convert multi-values categorical attribute into binary value space.

The mapping mechanism is intuitive. For example, $\mathbf{c}^j$ is the jth categorical attribute in $\mathcal{D}$. We assume $\mathbf{c}^j$ has K different categorical values, $\mathbf{c}^j \in \{a_i^{j,1}, a_i^{j,2}, \cdots, a_i^{j,K}\}$. Then each mapped categorical attribute value $\bar{c}_i^j$ is formed by a binary vector with the length of K. For example, $\mathbf{c}^j \in \{1, 2, 3\}$, $a_i^{j,1} = 1$, $a_i^{j,2} = 2$, $a_i^{j,3} = 3$, then their mapped attribute value should be $\bar{a}_i^{j,1} = [1, 0, 0]$, $\bar{a}_i^{j,2} = [0, 1, 0]$, $\bar{a}_i^{j,3} = [0, 0, 1]$, respectively. If $c_i^j = 1$, then $\bar{c}_i^j = [1, 0, 0]$

The mapped categorical attribute value $\bar{c}_i^j$ has K binary values with only one element equal to 1. The mapping algorithm is very simple, and there is no information loss. Therefore, any mixed attribute dataset with multi-value categorical attributes can be readily handled by our proposed algorithm below.

POD algorithm. Because most outlier detection approaches require fine-tuning of their parameters through trial-and-error approach, see e.g., KNN [7], which is impractical, because real world data usually do not contain labels for anomalous objects. Top-n style outlier detection methods alleviate the parameter setting problem somewhat. They provide a ranked list of objects that represent the degree of outlierness of each object. In POD, we employ top-n style outlier detection, which outputs the n objects with the highest MADOF values [7]. Algorithm 1 outlines the top-n style POD.

4 Experimental Results and Comparison

We now compare the outlier detection performance of top-n POD with LOADED [10] which is a typical mixed attribute outlier detection algorithm,

Algorithm 1. Top-n algorithm for POD

Input: A given dataset $\mathcal{D}$ pre-processed by the mapping mechanism presented at the beginning of Section 3, natural numbers n and k_{nn}.

1. For each subspace $\mathcal{S}'^k$;
   (a) learn parameter $\mathbf{w}^k$ for pattern P^k $(k = 1, 2, \cdots, N)$, based on Equation 2.
2. For each object $\mathbf{O}_i$ in $\mathcal{D}$
   (a) retrieve $\mathbf{O}_i$'s k-nearest neighbours by the neighbourhood size k_{nn};
   (b) calculate γ_i;
   (c) for each projected object $\mathbf{O}_i'^k$
      i. using the parameter $\mathbf{w}^k$ learnt, calculate β_i^k based on Equation 4;
   (d) calculate MADOF_i for each data object $\mathbf{O}_i$ based on Equation 5;
3. Sort the objects according to their MADOF values.
4. **Output:** the first n objects with the highest MADOF values.

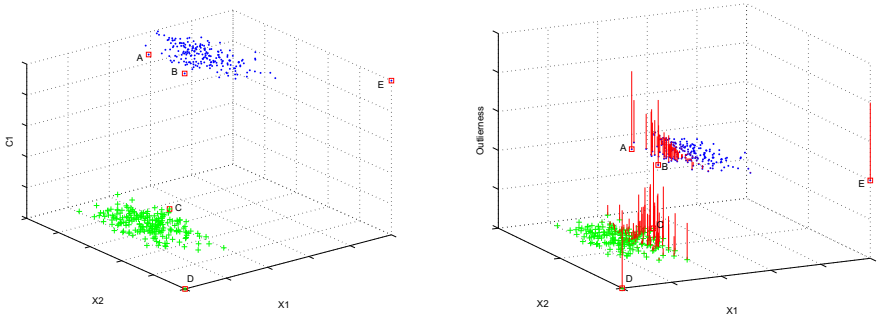
and two typical top-n methods, top-n LOF [15] and top-n KNN [14] which are used widely in real world applications. All the four algorithms were implemented in C++[1]. Our experiments start with a synthetic mixed attribute dataset which contains outliers that are meaningful but are difficult for the existing algorithms, LOADED, LOF and KNN. In Experiments 2, 3 and 4, we identify outliers from three real world mixed attribute datasets from the UCI machine learning data repository to illustrate the effectiveness of our method in real world situations. We recode categorical values and normalise all values so as to be used in KNN and LOF. For consistency, we only use the parameter k_{nn} to represent the neighbourhood size in the investigation of the methods used in our experiments. In all of our experiments, we set neighbourhood size k_{nn} equal to 1% of the cardinality of the observed dataset for POD, KNN and LOF, and set n equal to the number of genius outliers containing in the datasets. We use detection precision[2] to evaluate the performance of each method.

4.1 Synthetic Data

In Figure 2(a), there are 400 mixed attribute objects with two numerical attributes and one binary categorical attribute. All "cross" objects are assigned categorical value 0, while all "dot" objects have categorical value 1. There are five genuine outliers in the dataset, A, B, C, D and E. Most of outlier detection methods can identify D and E as outliers because they are typical under the homogenous data outlier definition, called traditional outliers [15]. However, A, B and C are special outliers in mixed attribute datasets. They are more likely to be in a group opposite to the group indicated by their categorical values. Therefore A, B and C are mixed attribute outliers, as they deviate from the pattern

[1] Source code of all algorithms used in experiments are available on the website: users.rsise.anu.edu.au/~kzhang

[2] Precision= $n_{\text{genius-outliers in top-n}}/n$. We set n as the number of genius outliers, and thus do not need to consider other outlier detection measures like recall.



(a) A 3-D synthetic dataset (b) MADOF (favorable view in colour)

Fig. 2. POD on a dataset with one categorical and two continuous attributes. MADOF values are proportional with the length of vertical line segments.

(normal behaviour) in the mixed attribute data. Figure 2(b) plots the MADOF values, which are proportional to the length of vertical line segments connected with these objects. Clearly, our method is able to detect both traditional outliers and mixed attribute outliers by considering interactions between categorical and numerical attributes. On the contrary, the other three methods are incapable of detecting A, B and C as these three points are not far away enough from their majorities. POD has 100% detection precision compared with 40% precision of the other three methods.

4.2 Real World Data

Adult data. The Adult dataset contains 48,842 data records with 6 numerical and 8 categorical attributes. The dataset was extracted from the US Census Bureau's Income dataset. Each record has features that characterise an individual's yearly income together with a class label indicating whether the person made more or less than 50,000 dollars per year. The first experiment on this dataset follows the same experiment setting described in [10]. Compared with results generated by LOADED [10], the results of POD seem more interesting. We list the top four outliers detected by POD:

- "A 90 years old male, working for local government for 40 hours per week with the job of craft repair and making more than 50,000 dollars per year."
- "A 73 years old self-employed female, working in clerical position for 99 hours per week and making less than 50,000 dollars per year."
- "A 61 years old self-employed female, working for 99 hours per week and making less than 50,000 dollars per year."
- "A 36 years old male, working in an executive position for 90 hours per week and making less than 50,000 dollars per year."

Table 1. Experimental and comparison results of POD over the three datasets. The detecting precision for each method based on 30 independent runs. The bold numbers indicate that the detection precision is statistically significantly improved compared to the other three methods (paired T-test at the 0.05 level).

Dataset	#Outliers	Precision (mean ± std.)			
		POD	LOADED	KNN	LOF
Adult	50	**0.682**±0.087	0.117±0.061	0.000±0.0000	0.012±0.0022
Census income	50	**0.731**±0.053	0.217±0.045	0.000±0.0000	0.049±0.0027
Housing	10	0.439±0.067	0.07±0.005	0.371±0.0024	0.000±0.0000

In order to make our experimental results comparable, we define the outlier label for the Adult data. We define that the data records with 80 or less work hours per week are normal, while above 80 hours per work are outliers. We independently run experiments 30 times for each of all the four algorithms, and in each time randomly pick up 50 outliers and 10,000 normal data. The average detection precisions and corresponding standard deviations of each method are listed in the Table 1. From the second row, we can see that the average detection precision for POD, LOADED, KNN and LOF is 68.2%, 11.7%, 0% and 1.2%, respectively. The paired T-test (at the confidence level, 0.05) indicates that there is statistically significant difference between POD and the other three methods.

Census income data. The census income (KDD) data contains weighted census data extracted from the 1994 and 1995 Current Population Surveys conducted by the U.S. Census Bureau. We use a part of the testing dataset which contains 99762 data records, with 7 numerical and 10 categorical attributes. We ignore the data records which are duplicated or have missing value (the pre-processed dataset is contained in our source code folder[1]). As the Census income dataset is similar with Adult dataset, we employ the same outlier label setting as in our Adult data experiment above. We also run experiments 30 times, and in each time randomly pick up 50 outliers and 10,000 normal data. As shown the third row in Table 1, we can see that the average detection precision for POD, LOADED, KNN and LOF is 73.1%, 21.7%, 0% and 4.9%, respectively. The paired T-test (at the confidence level, 0.05) indicates that there is statistically significant difference between POD and the other three methods.

Housing data. Housing data contains 506 data objects, with 14 attributes (11 numerical and 3 categorical). We use the fourth attribute, *Charles River dummy variable (1 if tract bounds river, 0 otherwise)* as the outlier label. We also run experiments 30 times, and in each time randomly pick up 10 outliers and mixed them with all the normal data. From the fourth row in Table 1, we can see that the average detection precision for POD, LOADED, KNN and LOF is 43.9%, 7%, 37.1% and 0%, respectively. The paired T-test indicates that there is statistically significant difference between POD and LOADED or LOF at the confidence level of 0.05, and between POD and KNN at the confidence level of 0.10.

5 Conclusions

In this paper, we have proposed a pattern based outlier detection approach, named POD, for mixed attribute data. Pattern is defined to describe majority of data as well as capture interactions among different types of attributes. In POD, the more does an object deviate from these patterns, the higher is its outlier factor. We have used logistic regression to learn patterns and formulate the outlier factor in mixed attribute datasets. Furthermore, a top-n style POD algorithm has been given and tested over both synthetic and real world datasets. Experimental results have demonstrated the effectiveness of top-n POD that can detect outliers in mixed attribute data statistically significantly better than LOADED, top-n KNN and top-n LOF. We are planning to investigate more complicated patterns.

Acknowledgement. The work was partially supported by the CSIRO-Centrelink Human Services Delivery Research Alliance.

References

1. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: Identifying density-based local outliers. In: SIGMOD, pp. 93–104 (2000)
2. Jin, H., de Vel, O., Zhang, K., Liu, N.: Knowledge discovery from honeypot data for monitoring malicious attacks. In: Wobcke, W., Zhang, M. (eds.) AI 2008. LNCS (LNAI), vol. 5360, pp. 470–481. Springer, Heidelberg (2008)
3. Jin, H., Chen, J., He, H., Kelman, C., McAullay, D., O'Keefe, C.M.: Signaling potential adverse drug reactions from administrative health databases. IEEE Transactions on Knowledge and Data Engineering 22(6), 839–853 (2010)
4. Barnett, V., Lewis, T.: Outliers in Statistical Data. John Wiley, New York (1994)
5. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Survey 41(3) (2009)
6. Knorr, E.M., Ng, R.T.: A unified notion of outliers: Properties and computation. In: KDD, pp. 219–222 (1997)
7. Zhang, K., Hutter, M., Jin, H.: A new local distance-based outlier detection approach for scattered real-world data. In: PAKDD, pp. 813–822 (2009)
8. Fan, H., Zaïane, O.R., Foss, A., Wu, J.: A nonparametric outlier detection for effectively discovering top-n outliers from engineering data. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) PAKDD 2006. LNCS (LNAI), vol. 3918, pp. 557–566. Springer, Heidelberg (2006)
9. Bay, S.D., Schwabacher, M.: Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: KDD, pp. 29–38 (2003)
10. Ghoting, A., Otey, M.E., Parthasarathy, S.: LOADED: Link-based outlier and anomaly detection in evolving data sets. In: ICDM, pp. 387–390 (2004)
11. Otey, M., Parthasarathy, S., Ghoting, A.: Fast lightweight outlier detection in mixed-attribute data. In: Techincal Report, OSU–CISRC–6/05–TR43 (2005)
12. Yu, J.X., Qian, W., Lu, H., Zhou, A.: Finding centric local outliers in categorical/numerical spaces. Knowl. Inf. Syst. 9(3), 309–338 (2006)
13. Ye, M., Li, X., Orlowska, M.E.: Projected outlier detection in high-dimensional mixed-attributes data set. Expert Syst. Appl. 36(3), 7104–7113 (2009)
14. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. In: SIGMOD, pp. 427–438 (2000)
15. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: Identifying density-based local outliers. In: SIGMOD Conference, pp. 93–104 (2000)

Incremental Projection Vector Machine: A One-Stage Learning Algorithm for High-Dimension Large-Sample Dataset

Qinghua Zheng[1], Xin Wang[1,3,*], Wanyu Deng[1,3,*], Jun Liu[1], and Xiyuan Wu[1]

[1] The MOE KLINNS Lab and SKLMS Lab, Department of Computer Science
Xi'an Jiaotong University, 710049, China
[2] Xi'an Institute of Posts & Telecommunications, 710121, China
[3] {xinwang.xjtu,wanyu.deng}@gmail.com

Abstract. Dimension reduction has been widely employed to deal with the curse of dimensionality before training supervised learning such as neural network and this framework combining dimension reduction and supervised learning algorithms is called as two-stage approach. However during the process of this approach, the system has to store original data and pre-process data simultaneously which will increase the complexity and re-compute the SVD when the new data arrive. To address the above problems, this paper proposes a novel learning algorithm for high-dimension large-scale data, by combining a new incremental dimension reduction with feed-forward neural network training simultaneously, called Incremental Projection Vector Machine (IPVM). With new samples arriving, instead of re-computing the full rank SVD of the whole dataset, an incremental method is applied to update the original SVD. It is suitable for high-dimension large-sample data for the singular vectors are updated incrementally. Experimental results showed that the proposed one-stage algorithm IPVM was faster than two-stage learning approach such as SVD+BP and SVD+ELM, and performed better than conventional supervised algorithms.

Keywords: Singular vector decomposition, Neural network, Extreme Learning Machine, Projection Vector Machine, Incremental Projection Vector Machine.

1 Introduction

High-dimension problems are often confronted in the process of data mining, which lower the efficacy of the conventional machine learning and data mining techniques greatly because of the so-called curse of dimensionality [8]. The general solution is to apply dimension reduction such as Singular Value Decomposition (SVD)[9], linear PCA[10], linear LDA[11], multidimensional scaling (MDS)[12], diffusion maps[14], multilayer autoencoders[13] and Laplacian Eigenmaps[15] as a pre-processing step, and then several data mining techniques can be used to learn the low dimensional data. And SVD is applied widely for its simplicity and high efficacy. However, at the

* Corresponding Tel.: +86 29 8266 5262 811

J. Li (Ed.): AI 2010, LNAI 6464, pp. 132–141, 2010.

first step of the two-stage approach, the intermediate results computed with SVD have to be stored, which will increase the space complexity significantly. Additionally, it will become difficult for developing on-line learning and on-line updating algorithm, for we have to re-compute the SVD when the new data arriving.

In order to avoid these disadvantages, Deng [1] proposed a novel one-stage learning algorithm called Projection Vector Machine (PVM) which combines SVD with single hidden layer feed-forward neural networks (SLFN) together. With SVD, the algorithm reduces the dimension and obtains the suitable size of feed-forward neural network and input weights simultaneously. Experimental results showed that the method performed better than conventional methods. According to Deng, the method currently is just suitable for high-dimension small-sample problem, when it comes to large-scale data, it is always lack of enough space to compute a full SVD which is fundamentally an O(PQ·min(P,Q))-time problem where P and Q are the numbers of rows and column of corresponding matrix. Additionally, more and more applications where data must be incorporated into the SVD as they arrive, such as many computer vision algorithms calling for a "running" thin SVD of a video stream, financial transaction streams and network activity stream as well. In the end the practical need for updating an SVD instead of re-computing SVD to incorporate new data has become more and more acute.

To address the above issues we propose a new learning method to improve the PVM method, that is, the whole training data are not applied to compute SVD in order to save enough space, instead, we divide the original training dataset into two parts: the initial dataset used to compute an initial SVD, and the updating dataset to update the initial SVD incrementally, then the final SVD is obtained to learn a SLFN in the way described in [1].With this thought we call our method **I**ncremental **P**rojection **V**ector **M**achine (**IPVM**). IPVM shares the same advantages as PVM; moreover, as the SVD can be computed incrementally, it is convenient to compute large-sample data without worrying about the shortage of memory space.

The rest of the paper is arranged as follows. Section 2 introduces the related work. The algorithm is detailed in section 3. In section 4, comparative experiments are done and results are given to show the performance. In the end, the final conclusions are drawn and future work is given in section 5.

2 Related Work

2.1 The SVD in Data Mining

The singular value decomposition decomposes a matrix X into three orthogonal matrices U , V and a diagonal matrix $S = diag(s)$ whose values on the diagonal are nonnegative and in descending order , such that $USV^T = X$ .The SVD has the optimal truncation property: The r largest singular values together with their corresponding singular vector elements, the product of the resulting matrices $U_R S V_R^T \approx X$ is the best rank-R approximation of X and captures the important "latent" structure of the dataset.

2.2 Single Hidden Layer Feed Forward Networks (SLFNs)

SLFNs[2] have been extensively used in many fields due to their abilities:(1) to approximate complex nonlinear mappings directly from the input samples; and (2) to provide models for a large class of natural and artificial phenomena that are difficult to handle with classical parametric techniques.

For N arbitrary distinct samples $\{(x_i, t_i)\}_{k=1}^{N}$, where $x_i \in [x_{i1}, x_{i2}...x_{iR}]$ and $t_i = [t_{i1}, t_{i2}, ..., t_{il}]$, standard SLFNs with $\tilde{N}$ hidden neurons and activation function $g(x)$ are mathematically modeled as $\sum_{i=1}^{\tilde{N}} \beta_i g(w_i, [x_k, b_i]) = t_k, k = 1, ..., N$ where $w_i = [w_{i1}, w_{i2}, ..., w_{i(R-1)}, w_{iR}]$ is the weight vector connecting the i-th hidden neuron and the input neurons, $\beta_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{il}]$ is the weight vector connecting the i-th hidden neuron and the output neurons, and b_i is the threshold of the i-th hidden node. $w_i.x_j$ denotes the inner product of w_i and x_j. These N equations can be written compactly as $H\beta = T$, where

$$H = \begin{bmatrix} g(w_1.x_1 + b_1) & ... & g(w_{\tilde{N}}.x_1 + b_{\tilde{N}}) \\ ... & ... & ... \\ g(w_1.x_N + b_1) & ... & g(w_{\tilde{N}}.x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}}, \beta = \begin{bmatrix} \beta_1 \\ ... \\ \beta_{\tilde{N}} \end{bmatrix}_{\tilde{N} \times l} \quad and \quad T = \begin{bmatrix} t_1 \\ ... \\ t_N \end{bmatrix}_{N \times M}$$

2.3 Extreme Learning Machine

To overcome the issues related with gradient-based algorithms such as the BP learning algorithm when training an Artificial Neuron Network (ANN), Huang et al. proposed a neural network training algorithm called the **E**xtreme **L**earning **M**achine (**ELM**) [2]. The input weights of an SLFN can be randomly chosen (any continuous distribution), and the output weights of an SLFN can be analytically determined by Moore-Penrose generalized pseudo-inverse [4]. ELM algorithm can be summarized as follows:

Given a training set X= $\{(x_i, t_i) \mid x_i \in R^R, t_i \in R^l, i = 1, 2, ..., N\}$, an activation function $g(x)$, and the number of hidden neurons $\tilde{N}$.

1. Randomly assign the input weights according to some continuous probability density function;
2. Calculate the hidden layer output matrix H;
3. Calculate the output weight β: $\beta = H^{+}T$.

2.4 Projection Vector Machine

The generalization of ELM depends on the proper selection of the fixed parameters, but it is difficult to find the best parameters ($\tilde{N}$,w_i) . In order to address the issue, Deng [1] proposed a novel algorithm called **P**rojection **V**ector **M**achine (**PVM**), where the projection from input layer to hidden layer by input weights $W_{m \times \tilde{N}}$ can be regarded as

dimension reduction $Z_{n\times\tilde{N}} = P_{n\times m}W_{m\times\tilde{N}}$ ($Z_{n\times\tilde{N}}$ is the intermediate matrix into hidden layer), and can be obtained from the projection matrix V_k by selecting the k singular vector elements corresponding to largest singular values directly. While the two-stage learning approach neglects V_k, and learns once again through ELM or BP in the second stage.. Based on this idea, the PVM can be described as follows:

Given a training set $X = \{(x_i, t_i) \mid x_i \in R^R, t_i \in R^l, i = 1,2,...,N\}$, an activation function $g(x)$.

1. Let $P = \{[x_k, 1] \mid x_k \in R^R\}_{k=1}^N, T = \{t_k \mid t_k \in R^l\}_{k=1}^N$;
2. Calculate SVD of P: $[U, S, V^T] = svd(P)$;
3. Set the number of hidden neurons $\tilde{N}$;
4. Obtain $w_i = V(:, 1:\tilde{N})$;
5. Calculate the hidden layer output matrix H;
6. Calculate the output weight $\beta: \beta = H^+T$.

---

3 The Proposed Algorithm: Incremental Projection Vector Machine

It is validated in [1] that when applied in most kinds of scenes, PVM has better accuracy compared with other algorithms. However, more and more applications require the real time performance especially with very large scale data. When we build a classification model with high-dimensional data, these high dimensions should be reduced in the preprocess step, and then the classification model is trained using these low-dimensional data. Once new data arrive, only the new data will be used to update the dimension-reduced data but not the original complete data, so that reduces the time complexity and saves a great amount of memory. With the above idea, we propose an algorithm called **I**ncremental **P**rojection **V**ector **M**achine (**IPVM**).

The Incremental Singular Value Decomposition can be specified to solve the following issue: Given several new column vectors C, what is the SVD of $[X\ C]$? Here we develop the special situation below:

Operation	Known	Desired	A	$\mathbf{B^T}$
Update	$US[V^T\ 0] = [X\ 0]$	$U_p S_p V_p^T = [X\ C]$	C	[0...0,1...1]

where X is the original matrix and C are the new coming samples. To add the new samples in the form of matrix, we firstly represent it as a new matrix which we denote as the product of A and B^T. Typically B is a binary vector indicating which columns should be modified, and A is derived from containing update (C).

Now the updating scenario can be described as: Given column vectors A, B and a known SVD $[U, S, V^T] = X$, what is the SVD of $X + AB^T$? Here we introduce a method called low-rank modifications of SVD proposed by M. Brand [3,5] and apply it in the scenario of the matrix's column updates.

Let $[U, S, V^T] = X$ with $U^TU = V^TV = I$ be a rank-r thin singular value decomposition of $X \in R^{M\times N}$. Let $A \in R^{M\times C}$, $B \in R^{N\times C}$ are arbitrary matrices of rank

C, the SVD of $X+AB^T$ (AB^T represents the additional part of the original matrix X) is shown in the next:

$$X+AB^T=[U\ P]K[V\ Q]^T \tag{1}$$

where P is an orthogonal basis of the column space of $(I-UU^T)A$ -the component of A orthogonal to U , and one can obtain it from the QR-decomposition. Similarly, let Q be an orthogonal basis of $B-VV^TB$. Therefore, K can be expressed as:

$$K=\begin{bmatrix} I & U^TA \\ 0 & R_A \end{bmatrix}\begin{bmatrix} S & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} I & V^TB \\ 0 & R_B \end{bmatrix}^T \tag{2}$$

The goal is to re-diagonalize K as $K=U'S'V'^T$, which gives rotation U' and V' of the extended subspaces $[U\ P]$ and $[V\ Q]$ such that

$$X+AB^T=([U,P]U')S'([V\ Q]V')^T \tag{3}$$

The detailed processes of IPVM can be described as follows:

Given a training set: X= $\{(x_k,t_k)\mid x_k\in R^M, t_k\in R^l, k=1,2,...,N\}$

1. **Initialization Phase**: Initialize the SVD using a small chunk of initial training data $X_1=\{\vec{x}_k\mid\vec{x}_k\in R^M\}_{k=1}^{N_1}$ from the given training dataset $X=\{\vec{x}_k\mid\vec{x}_k\in R^M\}_{k=1}^{N}$ which is applied to generate an initial SVD $\left[U,S,V^T\right]=SVD(X_1)$;
2. **Incremental Dimension Reducing Phase**: Updating U , S and V using low-rank modifications of SVD when the rest data of X come; $X_2=\{\vec{x}_k\mid\vec{x}_k\in R^M\}_{k=1}^{N_2}$ which is used to update the SVD incrementally above for several times as the new coming data,
   - i. $M=U^TA$; $p=A-UM$; $R_A=\|p\|$; $P=R_A^{-1}\cdot p$, where P is an orthogonal basis of the column-space of $(I-UU^T)A$, which is the component of " A " that is orthogonal to U ;
   - ii.Diagonalizing K and maintaining rank r: K is directly derived from (2);
   - iii.Decomposing K ,where $\left[U',S',V'\right]=SVD(K)$;
   - iv.Updating U , S and V to U_p , S_p and V_p : $U_p=[U\quad P]\cdot U'$; $V_p=V_p\cdot V'$ and $S_p=S'$.
3. **Model building**: Set the number of hidden neurons $\tilde{N}$
   - i. Obtain $w_i=V_P(:,1:\tilde{N})$;
   - ii. Calculate the hidden layer output matrix H;
   - iii. Calculate the output weight β : $\beta=H^+T$.

In the second part, instead of rotating the larger singular vector matrices U and V , we leave the SVD decomposed into matrices with orthonormal U , V , U' and V'. The large outer matrices only record the span of the left and right subspaces and are built by appending columns to U and rows to V . The transforms of these subspace bases are maintained in much smaller U' and V'; this makes the update much faster.

Remark1. For a high-dimensional low-rank matrix, the incremental SVD can be an effective linear-time algorithm and if the data is streamed through the CPU, the update requires less space to store the current SVD. That satisfies the demand of online updating.

Remark2. The IPVM can have better generalization performance than that of the gradient-based learning such as back-propagation when learning a SLFN model. For it can obtain the solutions straightforward without the trivial issues facing by the conventional gradient-based learning ones such as local minima, improper learning rate and over-fitting. On the other side, IPVM is more stable and accuracy than ELM.

4 Performance Evaluation

In this section, the performance including accuracy and time of the proposed IPVM are compared with ELM, ELM+SVD, BP, BP+SVD and PVM on different benchmark classification problems from UCI database[6], whose distributions are unknown and most of them are not noisy-free, in addition, all of them are high-dimensional. We utilize a fast BP algorithm called Levenberg-Marquardt algorithm in our simulations. The ELM's source code can be downloaded from Huang's homepage [7]. For each case, the training dataset and testing dataset are randomly generated from the whole dataset before each trial of simulation, and the specification of these benchmark problems are shown in Table 1.In our experiments, all the inputs(attributes) had been normalized into range [-1,1]. All the simulations were carried out in MATLAB 7 environment running in an Inter core 2, 2.66 GHZ CPU. In the end, the average results over 50 trials were obtained for these methods.

4.1 Selections of Parameters

First we selected the optimal parameters of all the methods. For ELM, ELM+SVD, BP, BP+SVD and PVM we chose the optimal number of hidden neurons by increasing its number at an interval of 5 with cross-validation method. For IPVM, the parameters contained the size of initial matrix, the block size used for updating and the optimal number of hidden neurons in the same way. Then the average results of 50 trials of simulations for each fixed size of SLFN were obtained and finally the best performance including training accuracy, testing accuracy, training time and testing time were discussed in this paper. To compare the total time of dimension reduction phase and model building phase, we computed the training time by adding the dimension reduction time and the model building time together in ELM+SVD and BP+SVD, and the time of three phases in IPVM. The final results are recorded in the next section.

Table 1. Specification of real-world classification cases

Dataset	#Observations		Attributes	
	Training	Testing	Continuous	Nominal
Madelon	2080	520	500	0
semeion	1274	319	1600	0
secom	1253	314	591	0
Sonar	166	42	60	0
face	160	240	1600	0
arcene	140	60	10000	0
Lung-cancer	25	7	56	0

4.2 Experimental Results and Analysis

IPVM works well for both one-by-one and chunk-by-chunk learning models. In this section, we only discuss the situation of chunk-by-chunk. To obtain the optimal performance we first adjusted the size of chunk depending on each problem. And in the next section we will study the relationships of the initial size, the updating block size and the performance of IPVM.

Experiments show that IPVM obtained higher generalization performance than ELM, BP, ELM+SVD, BP+SVD in general and was mostly faster than two-stage

Table 2. Training and testing accuracy of IPVM, ELM+SVD and ELM

Dataset	IPVM		ELM+SVD		BP+SVD	
	Training	Testing	Training	Testing	Training	Testing
semeion	0.8972±0.00468	0.8737±0.0170	0.9612±0.0225	0.8411±0.0184	-	-
Sonar	0.7980±0.0256	0.7648±0.0596	0.9004±0.0192	0.7395±0.0869	0.86280±0.0299	0.7576±0.069
Madelon	0.6214±0.0045	0.6045±0.0026	0.6146±0.0081	0.5528±0.0144	-	-
Lung-cancer	0.8768±0.0414	**0.7971**±0.1504	0.8768±0.0646	0.7200±0.1731	0.8488±0.0692	0.6771±0.1844
face	0.9772±0.0096	0.8579±0.0234	0.9463±0.0221	0.5398±0.0527	0.6062±0.1783	0.3352±0.0940
arcene	0.9543±0.0138	**0.8617**±0.0465	0.7761±0.0269	0.6383±0.0590	0.8346±0.1030	0.7333±0.0858
secom	0.9320±0.0039	**0.9400**±0.0142	0.9332±0.0025	0.9250±0.0098	0.9362±0.0146	0.9097±0.0358

Table 3. Training and testing accuracy of PVM, BP, BP+SVD

Dataset	PVM		BP		ELM	
	Training	Testing	Training	Testing	Training	Testing
semeion	0.8983±0.0051	0.8740±0.0139	-	-	0.9841±0.0033	0.8451±0.0178
Sonar	0.8229±0.0028	0.7681±0.0658	0.9004±0.0451	0.7395±0.0666	0.8978±0.0259	0.7324±0.0629
Madelon	0.6332±0.0049	0.6077±0.0023	-	-	0.6145±0.0097	0.5528±0.0225
Lung-cancer	0.8808±0.0453	0.7600±0.1427	0.8440±0.0590	0.6486±0.1853	0.8384±0.0599	0.7457±0.1588
face	0.9846±0.0102	0.8717±0.0296	-	-	0.9697±0.0164	0.4712±0.0604
arcene	0.9525±0.0134	0.8725±0.0462	-	-	0.8746±0.0269	0.5792±0.0757
secom	0.9346±0.0025	0.9303±0.0097	-	-	0.9250±0.0027	0.9333±0.0109

Table 4. Training and testing time (seconds) of IPVM, ELM+SVD and ELM

Dataset	IPVM		ELM+SVD		BP+SVD	
	Training	Testing	Training	Testing	Training	Testing
semeion	2.0456	0.0031	1.1160	0.0226	-	-
Sonar	0.1359	0.0006	0.0172	0.0016	0.9634	0.0100
Madelon	4.2681	0.0094	3.2481	0.0513	-	-
Lung-cancer	0.0321	0.0000	**0.0022**	0.0006	0.2588	0.0031
face	0.4141	0.0070	0.2125	0.01719	133.0266	0.0140
arcene	2.1438	0.0188	0.9797	0.0266	2.9195	0.0086
secom	**0.9008**	0.0039	1.9914	0.0351	20.3359	0.0078

Table 5. Training and testing time (seconds) of PVM, BP, BP+SVD

Dataset	PVM		BP		ELM	
	Training	Testing	Training	Testing	Training	Testing
semeion	**0.6219**	0.0056	-	-	1.1081	0.0184
Sonar	**0.0106**	0.0028	5.7344	0.0128	0.0113	0.0013
Madelon	**3.1534**	0.0088	-	-	3.2481	0.0512
Lung-cancer	**0.0038**	0.0000	2.0447	0.0075	0.0006	0.0000
face	**0.2039**	0.0117	-	-	0.0313	0.0109
arcene	**1.0063**	0.0188	-	-	0.3266	0.0227
secom	2.0695	0.0109	-	-	0.0250	0.0031

Table 6. The number of hidden neurons of all the algorithms

Dataset	IPVM	ELM	BP	ELM+SVD	BP+SVD	PVM
Sonar	**10**	50	10	50	10	15
Madelon	**8**	300	-	100	-	100
Lung-cancer	8	8	**6**	9	**6**	12
face	**30**	65	-	65	40	32
arcene	56	70	-	40	**8**	56
secom	12	30	-	**5**	**5**	42

learning algorithms. Table 2 and 3 present the testing accuracy of fix sized IPVM, PVM, ELM, BP, ELM+SVD and BP+SVD over 50 trials with different and same training set respectively. If the difference of the two testing accuracy obtained by two algorithms was larger than 0.005 for a case, we show the winner's testing accuracy in boldface. We can find that IPVM was more stable than other algorithms as well as PVM. This is because the input weights of IPVM are obtained from singular vectors of dataset by SVD while input weights of ELM are randomly assigned. Although BP determines its input weights by the gradient-based algorithm, the performance of BP are sensitive to initial values of input weights. This makes IPVM easy to obtain the optimal settings. In Table 6, although IPVM needed more hidden neurons than BP and BP+SVD, it was more compact than ELM and ELM+SVD in most cases. This means IPVM could have better responding ability than ELM and ELM+SVD. Table 4 and 5 display that the IPVM needed slightly longer training time than PVM and ELM did when datasets were small; while it was much shorter than that of BP.

The comparisons between IPVM and PVM show that the performance of them are similar to each other except that IPVM sometimes required more training time than PVM did and sometimes less. In next part, we will explain the reasons cause the phenomena.

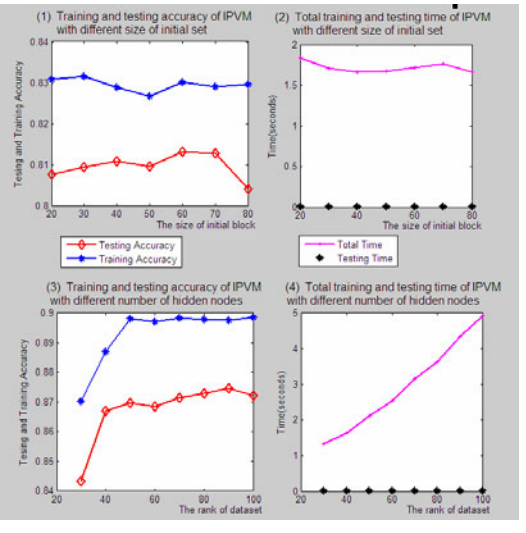


Fig. 1. Training ,testing accuracy and time of IPVM over 50 trials with different size of initial set (dataset:Semeion; block size: 50; the number of hidden neurons: 20; the size of initial set:100)

As the above figures shown, the conclusions can be obtained as following:

Remark A. The sizes of initial set and updating block both have little influence on the testing accuracy, training accuracy and time consummation (see Fig. 1.(1),(2)).

Remark B. Accuracy and time consumptions are closely related to the number of hidden neurons (see Fig. 1.(3) and (4)). They all increase when the number increases, and when it comes to a certain threshold, accuracy stops increasing. There is another thing we should pay attention to when utilizing IPVM. The number of hidden neurons must not over take the size of rank, for PVM, the input weight matrix $w_{m\times\tilde{N}}$ is seen as projection vectors mapping the data into low dimension, so the size of $\tilde{N}$ must be under the limitation of the rank. That means if the rank of a dataset is small, the model training phase can be fast. So IPVM is suitable for those datasets which have high dimension but low rank, and we can apply it to learn extreme large-scale data that PVM cannot do.

Remark C. Computing a full SVD is fundamentally an O(PQ×min(P,Q))-time problem, while the low-rank thin SVD of a $P\times Q$ matrix whose rank is r can be computed in O(PQr) time[3]. To obtain better performance, the initial datasets should be large enough to obtain a relatively large rank. However, with the increase of the rank size, it takes more time to carry out the incremental SVD. In the end, the total training time including the time of initial SVD and incremental SVD of IPVM sometimes overtakes the training time of PVM (see Table 4 and 5).

5 Conclusions and Future Work

In this paper, we propose a novel learning approach addressing the problems which PVM encounters: PVM is not suitable for large-scale datasets due to lack of memory, and has to update the singular vectors by retraining the whole dataset when new data arrive. When computing the dimension reduction result of one training dataset, we apply the low-rank modifications of SVD to reduce the high-dimensional data incrementally to get an SVD of the whole training dataset. Then we apply PVM to learn the new low dimensional dataset. Experimental results on UCI datasets showed that, IPVM had better performance than ELM, ELM+SVD and BP+SVD did in most cases. Moreover, it did not need iterative tuning input weight like BP and just needed only one-time learning like ELM. Compared with other two-stage learning algorithms, its training speed was faster. The last but not least is IPVM can be applied to train the extreme large-sample dataset without worrying about the shortage of memory, and it also can be employed in the online situation such as real-time personalized recommendation or dynamical user interesting reorganizations.

However, IPVM has some other problems particularly meriting our further study: It is only suitable for low-rank data, if the given dataset has a high rank, utilizing IPVM may not obtain a satisfying performance for the rank will be determined by the initial set whose size can significantly influent the final performance. Moreover, we have not updated the output matrix of hidden layer H incrementally when new data come, that means we have to re-compute H and β after carrying out the incremental SVD, which will lead to extra time and memory consummation.

Acknowledgement

The research was supported by the National High-Tech R&D Program of China under Grant No.2008AA01Z131, the National Science Foundation of China under Grant Nos.60825202, 60803079, 60921003, the National Key Technologies R&D Program of China under Grant Nos.2006BAK11B02, 2006BAJ07B06, the Program for New Century Excellent Talents in University of China under Grant No.NECT-08-0433 and the research on method of emotion interaction based on interatcive texts in e-learning under Grant No. 61070072.

References

1. Deng, W., Zheng, Q., Lian, S., Chen, L., Wang, X.: Projection Vector Machine: One-Stage Learning Algorithm from High-Dimension Small-Sample Data. In: Procceedings of the IEEE International Joint Conference on Neural Network, Barcelona, Spain, pp. 3375–3382 (2010)
2. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: theory and applications. Neurocomputing 70, 489–501 (2006)
3. Brand, M.: Fast Low-Rank Modifications of the Thin Singular Value Decomposition. Linear Algebra and Its Applications 415(1), 20–30 (2006)
4. Huang, G.-B.: Learning Capability and Storage Capacity of Two-Hidden-Layer Feedforward Networks. IEEE Transactions on Neural Networks 14(2), 274–281 (2003)
5. Brand, M.: Fast Online SVD Revisions for Lightweight Recommender Systems. In: SIAM International Conference on Data Mining (2003)
6. UCI database, `http://archive.ics.uci.edu/ml/`
7. ELM Source Codes, `http://www.ntu.edu.sg/home/egbhuang/`
8. van der Maaten, L.J.P., Postma, E.O., van den Herik, H.J.: Dimensionality Reduction: A comparative Review, `http://www.iai.uni-bonn.de/~jz/dimensionality_reduction_a_comparative_review.pdf`
9. Golub, G.H., Van Loan, C.F.: Matrix Computations. Johns Hopkins Univ. Press, Baltimore (1989)
10. Hotelling, H.: Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology 24, 417–441 (1933)
11. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics 7, 179–188 (1936)
12. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290(5500), 2319–2323 (2000)
13. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (2006)
14. Lafon, S., Lee, A.B.: Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(9), 1393–1403 (2006)
15. Belkin, M., Niyogi, P.: Laplacian Eigenmaps and spectral techniques for embedding and clustering. In: Advances in Neural Information Processing Systems, vol. 14, pp. 585–591 (2002)

Model-Based Viewpoint Invariant Human Activity Recognition from Uncalibrated Monocular Video Sequence

Zaw Zaw Htike[1], Simon Egerton[1], and Ye Chow Kuang[2]

[1] School of Information Technology
[2] School of Engineering
Monash University,
Sunway Campus, Malaysia
{zaw,simon.egerton}@infotech.monash.edu.my,
kuang.ye.chow@eng.monash.edu.my

Abstract. There is growing interest in human activity recognition systems, motivated by their numerous promising applications in many domains. Despite much progress, most researchers have narrowed the problem towards fixed camera viewpoint owing to inherent difficulty to train their systems across all possible viewpoints. Fixed viewpoint systems are impractical in real scenarios. Therefore, we attempt to relax the fixed viewpoint assumption and present a novel and simple framework to recognize and classify human activities from uncalibrated monocular video source from any viewpoint. The proposed framework comprises two stages: 3D human pose estimation and human activity recognition. In the pose estimation stage, we estimate 3D human pose by a simple search-based and tracking-based technique. In the activity recognition stage, we use Nearest Neighbor, with Dynamic Time Warping as a distance measure, to classify multivariate time series which emanate from streams of pose vectors from multiple video frames. We have performed some experiments to evaluate the accuracy of the two stages separately. The encouraging experimental results demonstrate the effectiveness of our framework.

Keywords: Viewpoint invariant, human activity recognition, 3D human pose estimation, Dynamic Time Warping.

1 Introduction

Human activity recognition is the recovery of human motion information from image sequences and labeling of the underlying activities of the human subjects. The problem of automatic human activity recognition has become very popular due to its countless promising applications in many domains such as video surveillance, video indexing, computer animation, automatic sports commentary systems, human computer interaction systems, context-aware pervasive systems, smart home systems and other human-centered intelligent systems. There are a number of reasons why human activity recognition is a very challenging problem. Firstly, a human body is

J. Li (Ed.): AI 2010, LNAI 6464, pp. 142–152, 2010.

non-rigid and has many degrees of freedom, generating infinitesimal variations in every basic movement. Secondly, no two persons are identical in terms of body shape, volume and coordination of muscle contractions, making each person generate unique movements. The above mentioned problems get further compounded by uncertainties such as variation in viewpoint, illumination, shadow, self-occlusion, deformation, noise, clothing and so on. Since the problem is very vast, it is customary for researchers to make a set of assumptions to make the problem more tractable. However, the most common and the biggest assumption made by researchers happen to be the 'fixed viewpoint assumption'. Their systems can recognize activities only from the 'trained' viewpoint. Unfortunately, the fixed viewpoint assumption is not valid in many domains. In video indexing, for example, viewpoint is arbitrary and may not even be stationary. In video surveillance, camera position is again arbitrary. That assumption is causing 'bottleneck' in practical applications [1]. Therefore, the *fixed viewpoint assumption* needs to be removed. We will therefore relax that assumption and present a simple and novel framework to recognize and classify human activities.

Fig. 1. Images resulting from variations in camera orientation

1.1 Related Work

There is indeed some confusion among researchers about the term *viewpoint invariance*. Some researchers such as [2] claim 'viewpoint invariance' when their system is only capable of recognizing sequences up to 45° horizontal deviation from frontal view. *Robustness* to some degree of variation in viewpoint is not the same as viewpoint invariance. A camera has 6 *degrees of freedom* (DOF). Viewpoint invariance refers to the ability of the system to produce consistent results wherever the camera is positioned and however it is orientated as shown in Fig 1, be it front-view, side-view, back-view or any intermediate view. In recent literature, there are mainly two branches of research that attack the viewpoint invariance issue: multiple-camera branch and single-camera branch. In a multiple-camera system, 3D information can be recovered by means of triangulation [3]. Some researchers fuse spatial information from multiple cameras to form what is called a *3D visual hull* [4-5]. Multiple-camera approach is the most widely investigated approach. Unfortunately, in many domains, applications are limited to single camera. For example, in video indexing, there are no data available from extra cameras. Single-camera approach is significantly more difficult than multi-camera approach [1; 6]. 100% viewpoint invariance has barely been achieved in the single-camera branch. Most of the recent single-camera techniques (for instance [7-8]) are still at best partially invariant to viewpoint. Thus we will focus only on the single-camera or monocular branch. Most single-camera approaches in the literature further branch into two major categories: **model-based approach** and **model-less approach**.

A model-based approach, which employs an explicit parametric anthropometric prior and attempts to recover structural information of the human body, is the more investigated approach. A human pose is represented by a kinematic tree model or a stick figure, consisting of joints linked by segments. Most of the existing works in the literature in the model-based branch concentrate on a lower-level field of research called 'pose recovery' rather than higher-level activity recognition because human pose recovery (which is a prerequisite to activity recognition) itself is an unsolved problem. A model-based approach estimates human pose either by direct inverse kinematics or by numerical optimization over the pose variables [9]. The two major problems that arise in a single-camera system are depth ambiguity and self-occlusion. Depth ambiguity arises because we are trying to reconstruct 3D skeleton out of 2D information. Recovering 3D information from a single uncalibrated camera is inherently ill-posed because we are trying to solve equations with more unknowns than the number of equations. Researchers try to disambiguate by bringing in more assumptions or constraints in one form or another. For example, Wei and Chai [10] use at least 5 key frames to resolve ambiguity and a numerical constrained optimization algorithm to construct 3D human poses. However, the system is still not so invariant to viewpoint because it does not work for top view. Shen and Foroosh [11] model a sequence of poses as a sequence of planes defined by triplets of body points. Despite good results, it cannot handle self-occlusion.

A model-less approach makes no attempt to recover structural information of the human body. Most model-less approaches such as [9] are example-based, that is they utilize machine learning techniques to construct a mapping function between 2D image features and 3D poses. Some researchers such as [7] find a common lower dimensional representation of the projected image of the same pose under different viewpoints. However, these kinds of approaches have not been demonstrated to be able to handle self-occlusion. The main disadvantage of model-less approaches is that the training examples should be very diverse and numerous so as to correctly map unknown poses. They are also generally more prone to overfitting.

1.2 Contributions

This paper has two major contributions: viewpoint invariant 3D human pose estimation and viewpoint invariant human activity recognition. We follow the model-based route. However, we show that full body pose recovery is not necessary to recognize activities. Unlike previous work in the activity recognition literature, we present a robust technique that can recognize activities from partial joint information such as when half of the body is missing. To be invariant to viewpoint, the system needs to be able to function correctly even with 'crippled' input. We demonstrate how our activity recognition system achieves full invariance to viewpoint under 6 DOF of camera.

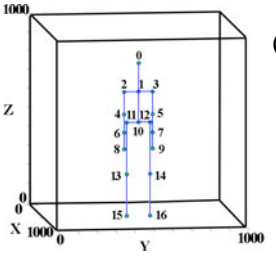
The paper is organized as follows. First, in Section 2, we explain how we estimate 3D human pose from a given video frame. In Section 3, we present how we extract pose vector from 3D human pose, merge pose vectors across successive frames to form a multivariate time series and then classify activities. We describe our experiments in Section 4 and conclude in Section 5.

2 Human Pose Estimation

First we need to estimate 3D human pose from each video frame. We employ a kinematic tree model of human body, consisting of 17 nodes or joints linked by 16 segments. Each joint has 3 DOF and predefined rotational ROM (range of motion). Estimating 3D pose from a 2D input image requires a list of approximate 2D coordinates of the joints as a prerequisite. The list can be obtained by body part detection algorithms. As body part detection itself is a diverse field of research, it will not be elaborated here due to space constraints. Further discussions can be found in [12]. We shall assume that body part detection has already been performed on the video frames, and that we are given an array of approximate 2D coordinates of 17 joints extracted from each frame as shown in Fig 3a. The input to our system is a vector $\mathbf{x} \in \mathbb{R}^{34}$ (Note that some of its components might be undefined for occluded joints). The pose estimation step takes $\mathbf{x}$ and produces a vector $\mathbf{y} \in \mathbb{R}^{51}$ which contains 3D coordinates of 17 joints. Since $\mathbf{x}$ can map to multiple $\mathbf{y}$, the previous output is used to disambiguate the mapping. The output of the estimation step for the i^{th} time step is then defined as $\mathbf{y}_i = f(\mathbf{y}_{i-1}, \mathbf{x})$, where $\mathbf{y}_{i-1}$ is the output of the previous time step. As there is no previous output for the first frame of a sequence, $\mathbf{y}_0$ is estimated through a lookup table which stores $\hat{\mathbf{x}}$-to-$\hat{\mathbf{y}}$ mappings of 50 primitive poses from 13 viewpoints. $\mathbf{y}_0$ is chosen from the lookup table as the value of $\hat{\mathbf{y}}$ in the table corresponding to $\hat{\mathbf{x}}$ that has the shortest Euclidean distance from the input $\mathbf{x}$.

Because of that fact that no two persons are identical in terms of body shape and volume and that skeleton size plays no role in activity recognition, we normalize the 3D human skeleton in a bounding cube (1000 units in each axis) in a right-handed coordinate system as shown in Fig 2a. Each segment has fixed length constraints as given in Fig 2b. Length constraints minimize the influence of inter-person structural differences.

(a)

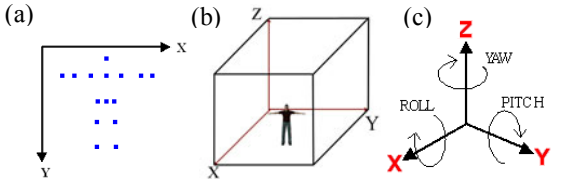


(b)

Segment	Distance	Segment	Distance
1-0	151	7-9	86
1-2	75	10-1	165
1-3	75	10-11	62
2-4	119	10-12	62
3-5	119	12-14	274
4-6	97	11-13	274
5-7	97	13-15	220
6-8	86	14-16	220

Fig. 2. (a) Normalized skeleton in 1000^3 bounding cube (b) Joint constraints table

A camera has 6 DOF with respect to an observed person: translations along x, y, z and rotations about x (roll, φ), y (pitch, θ), z (yaw, ψ). Fig 3c shows rotations about the axes. Because of normalization, translations along the axes have no effect on the system. Hence, our camera parameters are just ψ, θ and φ, each of which ranges from 0º to 360º. We standardize the direction of rotations as clockwise (following left-hand grip rule). Fig 3d shows rotational coordinates of basic viewpoints. Since the order of rotation matters in 3D, we will always follow the yaw-pitch-roll order.



(d)

Viewpoint	ψ	θ	φ
Front	0	0	0
Back	180	0	0
Right	270	0	0
Left	90	0	0
Top	0	90	0

Fig. 3. (a) Axes of the original data from preprocessing (b) Right-handed coordinate system of the 3D model (c) Yaw, pitch and roll (d) Rotational coordinates for various viewpoints

Fig 3a depicts an example of an original list of 2D coordinates of the joints obtained from preprocessing, with a negatively oriented coordinate system. If we assume for the moment that ψ, θ and φ are known, then projection equations given in Table 1 can be used to map **x** to a plane in the bounding cube. After that, we need to estimate 'depth'. For each segment, the ratio between the length of the segment and the average length of all the segments is calculated. The intuition is that if the ratio of a particular segment is smaller than that of the standard model, the segment must have some depth component whose direction is defined by the unit vector of the camera's line of sight (obtainable from ψ, θ and φ). Whether to project the depth component into the positive or negative direction of the unit vector depends upon the 'legality' of the new pose as defined by the joint rotational constraints. If both directions are allowed, multiple outputs will be produced. After all iterations of depth estimation, we normalize each segment's length. We keep the slope of each bone constant and change its length to that of the standard model defined in Fig 2b.

Table 1. Projection equations

	X	Y	Z
Roll (φ)	500	Sin(φ) x Y + Cos(φ) x X	- Cos(φ) Y + Sin(φ) X
Pitch (θ)	Cos(θ) X – Sin(θ) Z	Y	Sin(θ) X + Cos(θ) Z
Yaw (ψ)	Cos(ψ) X + Sin(ψ) Y	Sin(ψ) X + Cos(ψ) Y	Z

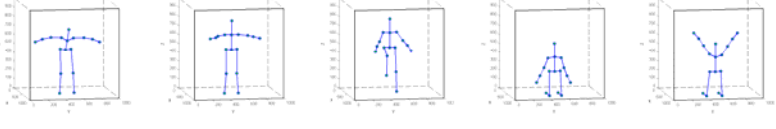


Fig. 4. 3D stick figures corresponding to various poses

After centering the skeleton in the cube with feet in the x-y plane, we get a 3D skeleton as shown in Fig 4. So far, we have assumed that ψ, θ and φ are known. But in reality we do not know the orientation of the camera. However, since inter-frame changes in relative camera position can be very small, ψ, θ and φ will not be so much different from the values of the preceding frame. We perform an exhaustive parallel search over ψ, θ and φ with initial values of the preceding frame bounded by ±45° envelope with 5° discrete step size. The search generates a list of legal 3D poses with ±5° tolerance. The last step of the pose estimation step is to choose one pose **y** from

the list that is most similar to previous output $\mathbf{y}_{i-1}$. Out of all legal poses in the list, we choose the one with shortest Euclidean distance from $\mathbf{y}_{i-1}$.

3 Human Activity Recognition

Activity recognition is the step after pose estimation. As the estimated pose is 3D, the resulting activity recognition system is inherently invariant to viewpoint. First, we extract a relevant *pose vector* from the 3D pose configuration. Each joint has 3 DOF. Fig 5 illustrates the degrees of freedom of right elbow. Note that the third DOF 'roll' or 'twist' is redundant for most joints in the stick figure pose representation. We represent pose vector by the configuration of the 12 joints as shown in Fig 6a. Each joint is represented by 2 angles (yaw and pitch). Therefore, each pose is represented by a 24-dimensional vector **p**. An activity is a sequence of poses. An activity is, therefore, represented by a multivariate time series matrix comprising 24 columns. Fig 6b illustrates one particular column of the matrix for the activity 'jumping'.

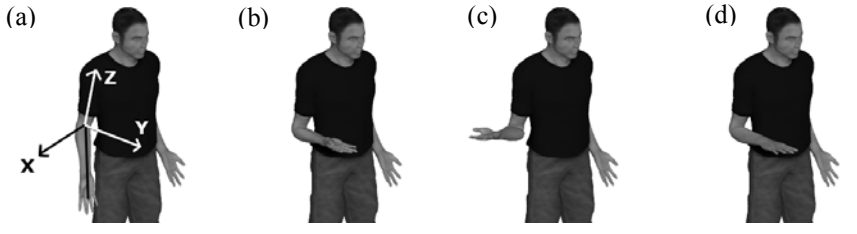


Fig. 5. (a) Original orientation of the forearm (b) Changing pitch angle from 0° to 90° (c) Changing yaw angle to 90° from b (d) Changing roll angle to 180° from b (twisting forearm)

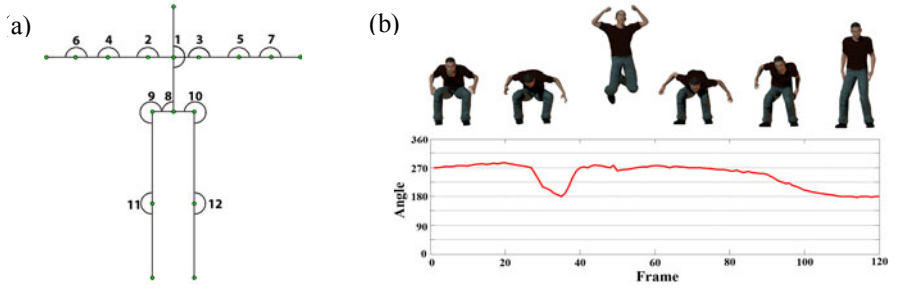


Fig. 6. (a) The 12 joints to present pose vector (b) a univariate time series comprising pitch angle of the right knee during the activity '*jumping*'

We classify activities using Nearest Neighbor Algorithm (NN) with Dynamic Time Warping (DTW) is a distance measure. Dynamic Time Warping (DTW) is a well-known algorithm for time series comparison in the literature. DTW minimizes the effect of time shifting, distortion and scaling [13]. Uniform scaling is a desired property in activity recognition due to inherent spatial and temporal variability found

in human motion. For example, a person may walk slowly or swiftly. Due to space constraints, we will not describe DTW in detail. Interested readers can refer to [14-15]. The only significant drawback of standard DTW is its relatively high computational cost $O(n^2)$ for comparing sequences of length n. However with global constraints (such as Sakoe-Chiba Band[16] and Itakura Parallelogram[17]) and various lower-bounding techniques (such as LB_Keogh [14]), the complexity can be reduced to almost linear time. In [14] , LB_Keogh lower-bounding technique, however, works only for univariate time series. For activity recognition, we extend Keogh's technique to perform lower-bounding of multivariate time series just like in [18]. The proof of the lower-bounding property of multivariate time series is also presented in [18]. DTW is essentially a global distance measure between two time series. DTW needs a local distance measure between two static points in the two time series. In the case of univariate time series, the local distance, d, between any two points in the time series, is simply the square-difference. For example, $d(3,4) = (3-4)^2$. For our multivariate case, the local distance, d, is the Euclidean distance between the two pose vectors.

$$d(\boldsymbol{a}, \boldsymbol{b}) = \sum_{i=1}^{N} (\boldsymbol{a}[i] - \boldsymbol{b}[i])^2 \quad (1)$$

where N is the dimension of the multivariate time series. Fig 7 depicts DTW of univariate time series and multivariate time series. The best thing about our algorithm is that N is adjustable based on the availability of joint information. For example, Fig 8

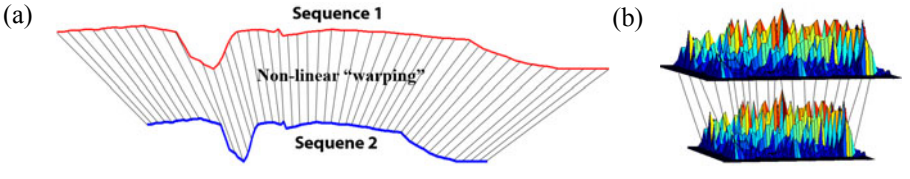


Fig. 7. (a) DTW for univariate time series) (b) DTW for multivariate time series

(b)

```
double LowerBound1NN(Sequence input, out int match_index)
{
            double closest = double.PositiveInfinity;
            int i = 0;
            foreach (Sequence sequence in database)
            {
                LB_distance = LowerBoundDTW(input, sequence);
                if (LB_distance < closest)
                {
                    double distance = DTW(input, sequence);
                    if (distance < closest)
                    {
                        closest = distance;
                        match_index = i;
                    }

                }
                i++;
            }
           return i;
}
```

Fig. 8. (a) "Jumping jacks" seen from right (b) Pseudo-code of 1-NN DTW sequential search algorithm with lower-bounding

shows part of a right-view image sequence of a person doing "jumping jacks" where left arm and leg are fully occluded. Unlike other activity recognition systems in the literature, our algorithm can leave out all the missing joints and compute d and DTW only based on available information. This makes our system very robust.

As a typical NN algorithm, there is no specific learning phase. Our system stores a list of multivariate time series of known activities and their corresponding labels in a database. When an unknown activity is presented to the system, the system takes the unknown time series, performs a sequential search with lower-bounding (as shown in Fig 8b) and outputs the label of the known activity which has the shortest global distance from the unknown time series. The system is scalable and suitable to be employed in domains such as video indexing.

4 Experiments

We carried out two separate experiments to evaluate pose estimation performance and activity recognition performance. For pose estimation, we used two of our own datasets. The first dataset comprised 200 static poses from 5 viewpoints (front, back, left, right and top) generated by *POSER PRO* [19]. The poses were taken from the library that came with the software package. So the first dataset contained 1000 static poses in total. Fig 9 depicts some of the poses from our dataset. The second dataset contained poses synthesized from 2 motion sequences (jumping and walking) with 120 frames for each sequence taken from 5 viewpoints. So the second dataset contained a total of 1200 poses. The ground truths for all the poses were obtained by a *Python script* that translated POSER's coordinate system to ours. For each pose, the ground truth was the 3D coordinates of all the joints. To evaluate activity recognition, we used CMU Motion-Capture database [20]. In fact, there were well-known datasets for viewpoint invariant human activity recognition such as IXMAS dataset [4]. However, since those standard datasets contained no annotated joint information, the CMU dataset (which provides 3D joint information) was our only choice. We selected 10 activities (dribbling, walking, running, jumping, boxing, dancing, waving, sitting,

Fig. 9. Our dataset to test pose estimation

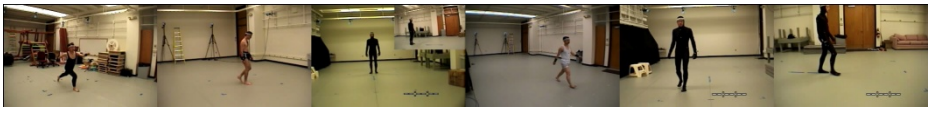


Fig. 10. CMU dataset to test activity recognition

climbing and kicking ball). Each activity was performed 4 times (mostly by the same actor). Fig 10 depicts some of the scenes from the CMU dataset. Their skeleton model was mapped to ours and joints coordinates are converted accordingly.

For pose estimation test, we took the *symmetric mean absolute percentage error* (SMAPE) as an error measure [21].

$$SMAPE = \frac{100}{n} \sum_{i=1}^{n} \left(\frac{||\boldsymbol{p}_i| - |\boldsymbol{t}_i||}{|\boldsymbol{p}_i| + |\boldsymbol{t}_i|} \right) \quad (2)$$

where n is the total number of poses, $\mathbf{p}_i$ is the i^{th} estimated pose vector and $\mathbf{t}_i$ is the i^{th} ground truth pose vector. Note that the default dimension of $\mathbf{p}_i$ and $\mathbf{t}_i$ is 24. But some components of $\mathbf{p}_i$ might be undefined due to occlusion. In those cases, we reduced the dimensionality by neglecting undefined components in $\mathbf{p}_i$ and the corresponding ones in $\mathbf{t}_i$. In activity recognition tests, we performed leave-3-out cross-validation. For each activity, we trained the system on the data from 1 out of 4 clips and then tested the system using the data from the other 3 clips. Since there were 4 combinations of picking one clip for training, the whole process was repeated 4 times and the resulting values were averaged. To test the robustness of our activity recognition system, we tested on various values of N (number of joints in the pose vector). We respectively tested without wrists, without lower body and without arms and shoulders. Table 2 and 3 list our experimental results. The results for Table 2 were calculated using (2) where n is 200 for static poses and 120 for motion sequences, for each viewpoint. Note that Table 2 reports error rates whereas Table 3 reports accuracy rates.

Table 2. Pose estimation results

	static poses	sequences
Front view	14.7%	12.3%
Back view	15.0%	13.8%
Left view	16.5%	13.2%
Right view	16.2%	13.9%
Top view	17.1%	14.1%
Average	15.9%	13.5%

Table 3. Activity recognition results

	Accuracy
N=12	97.5%
N=10 (without wrists [joints 6 and 7]*)	97.5%
N=8 (without lower body [9 to 12])	35.8%
N=6 (without arms & shoulders [2 to 8])	80.0%

*Note: Excluded joints numbers, according to Fig 6

The results of the post estimation tests demonstrate that the proposed system achieves decent performance in pose estimation. The error rates have shown to decrease when estimating poses from motion sequences. Despite one-shot learning, the results of the activity recognition tests demonstrate that our system easily achieves results on-par with current state of the art fixed view methods. The fact that the second test (N=10) gave the same accuracy rate as the first test (N=12) implies that wrist movement is minimal in the dataset and that it is redundant to take wrist configuration. The third test (N=8) gave very low accuracy rate (which was expected) because almost all the activities (especially running and kicking) had the highest variance in lower body configuration. Finally, the last test (N=6) did not produce low accuracy rate because only a few activities had the highest variance in arm configuration.

5 Conclusion and Future Work

We have presented a novel approach to viewpoint invariant human activity recognition system from uncalibrated monocular video source. Our system can learn from a small set of training examples. Our analysis and experiments show that we can indeed achieve viewpoint invariance in human activity recognition with high accuracy. This prototype limits classification of human activities to just 10 classes under a closed world assumption, but there are countless real-world activities. Since our system is scalable and the test results are promising, we could extend further to recognise a variety of common human activities.

As future work, we would first like to select a suitable body part detection algorithm from the literature and plug into our system. We would then obtain a standalone activity recognition system and be able to test our system on a variety of datasets.

References

1. Ji, X., Liu, H.: Advances in View-Invariant Human Motion Analysis: A Review. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 40(1), 13–24 (2010)
2. Holte, M.B., Moeslund, T.B.: View invariant gesture recognition using 3D motion primitives. Paper Presented at the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2008 (March 31-April 4 2008)
3. Yung-Tai, H., Jun-Wei, H., Hai-Feng, K., Liao, H.Y.M.: Human Behavior Analysis Using Deformable Triangulations. Paper Presented at the 2005 IEEE 7th Workshop on Multimedia Signal Processing (October 30-November 2, 2005)
4. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. Comput. Vis. Image Underst. 104(2), 249–257 (2006)
5. Jin, N., Mokhtarian, F.: Image-based shape model for view-invariant human motion recognition. Paper Presented at the IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS 2007 (September 5-7, 2007)
6. Sminchisescu, C.: 3D Human Motion Analysis in Monocular Video Techniques and Challenges. In: Proceedings of the IEEE International Conference on Video and Signal Based Surveillance, p. 76. IEEE Computer Society, Los Alamitos (2006)
7. Souvenir, R., Babbs, J.: Learning the viewpoint manifold for action recognition. Paper Presented at the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008 (June 23-28, 2008)
8. Yeyin, Z., Kaiqi, H., Yongzhen, H., Tieniu, T.: View-invariant action recognition using cross ratios across frames. Paper Presented at the 16th IEEE International Conference on Image Processing (ICIP) (November 7-10, 2009)
9. Agarwal, A., Triggs, B.: Recovering 3D human pose from monocular images. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(1), 44–58 (2006)
10. Wei, X.K., Chai, J.: Modeling 3D Human Poses from Uncalibrated Monocular Images. In: 12th IEEE International Conference on Computer Vision, Kyoto, Japan (2009)
11. Shen, Y., Foroosh, H.: View-Invariant Action Recognition from Point Triplets. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(10), 1898–1905 (2009)

12. Lee, M.W., Cohen, I.: Human body tracking with auxiliary measurements. Paper Presented at the AMFG 2003. IEEE International Workshop on Analysis and Modeling of Faces and Gestures (October 17, 2003)
13. Senin, P.: Dynamic Time Warping Algorithm Review, Honolulu, USA (2008)
14. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. Knowl. Inf. Syst. 7(3), 358–386 (2005)
15. Yi, B.-K., Jagadish, H.V., Faloutsos, C.: Efficient Retrieval of Similar Time Sequences Under Time Warping. In: Proceedings of the Fourteenth International Conference on Data Engineering, pp. 201–208. IEEE Computer Society, Los Alamitos (1998)
16. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. In: Readings in Speech Recognition, pp. 159–165. Morgan Kaufmann Publishers Inc., San Francisco (1990)
17. Itakura, F.: Minimum prediction residual principle applied to speech recognition. IEEE Transactions on Acoustics, Speech and Signal Processing 23(1), 67–72 (1975)
18. Rath, T.M., Manmatha, R.: Lower-Bounding of Dynamic Time Warping Distances for Multivariate Time Series. University of Massachusetts, Massachusetts (2003)
19. Pose Pro. 2010, Smith Micro (2010)
20. CMU Motion Capture Database, `http://mocap.cs.cmu.edu/`
21. Flores, B.E.: A pragmatic view of accuracy measurement in forecasting. Omega 14(2), 93–98 (1986)

On Optimizing *Locally* Linear Nearest Neighbour Reconstructions Using Prototype Reduction Schemes*

Sang-Woon Kim[1] and B. John Oommen[2]

[1] *Senior Member, IEEE*. Dept. of Computer Science and Engineering, Myongji University, Yongin, 449-728 South Korea
kimsw@mju.ac.kr

[2] *Chancellor's Professor* ; *Fellow : IEEE* and *Fellow : IAPR*. School of Computer Science, Carleton University, Ottawa, Canada : K1S 5B6. Also an Adjunct Professor with the University of Agder in Grimstad, Norway
oommen@scs.carleton.ca

Abstract. This paper concerns the use of Prototype Reduction Schemes (PRS) to optimize the computations involved in typical k-Nearest Neighbor (k-NN) rules. These rules have been successfully used for decades in statistical Pattern Recognition (PR) applications, and have numerous applications because of their known error bounds. For a given data point of unknown identity, the k-NN possesses the phenomenon that it combines the information about the samples from *a priori* target classes (values) of selected neighbors to, for example, predict the target class of the tested sample. Recently, an implementation of the k-NN, named as the Locally Linear Reconstruction (LLR) [11], has been proposed. The salient feature of the latter is that by invoking a quadratic optimization process, it is capable of systematically setting model parameters, such as the number of neighbors (specified by the parameter, k) and the weights. However, the LLR takes more time than other conventional methods when it has to be applied to classification tasks. To overcome this problem, we propose a strategy of using a PRS to efficiently compute the optimization problem. In this paper, we demonstrate, first of all, that by completely discarding the points not included by the PRS, we can obtain a reduced set of sample points, using which, in turn, the quadratic optimization problem can be computed far more expediently. The values of the corresponding indices are comparable to those obtained with the original training set (i.e., the one which considers all the data points) even though the computations required to obtain the prototypes and the corresponding classification accuracies are noticeably less. The proposed method has been tested on artificial and real-life data sets, and the results obtained are very promising, and has potential in PR applications.

1 Introduction

It is well known that the optimal classifier is the one that invokes the Bayes decision rule. If the *a priori* density functions were easily computable, and the class conditional

* The second author was partially supported by NSERC, the Natural Sciences and Engineering Research Council of Canada. This work was generously supported by the National Research Foundation of Korea funded by the Korean Government (NRF-2010-0015829).

J. Li (Ed.): AI 2010, LNAI 6464, pp. 153–163, 2010.

densities were truly of a classical well-defined nature (for example, of the exponential family), the tasks of training and testing a pattern recognition/classification system would be trivial. In practice, however, these distributions are far from ideal, and consequently, the science and art of PR has had to develop various non-parametric methods for training and testing. The most elementary of these, and yet the most well-developed, constitute the Nearest Neighbor (NN) family of classifiers[1].

The idea behind the NN rules is age-old and is essentially encapsulated in the axiom that the information about a particular sample point can be gleaned from its nearest neighbors. Traditionally, the consequent decision rule merely performs a majority decision based on the decision of the closest k neighbors. The beauty of such a scheme is that the decision rule asymptotically attains the accuracy of the Bayes rule as the number of neighbors, k, is increased. More recently, to yield even more accurate results (for any given value of k), researchers have proposed that the neighbors need not be assigned equal weights. Rather, the question is that of modeling every feature point as a convex combination of its k neighbors, and from this perspective, the crucial question is that of determining the weights that are to be assigned to these neighbors.

The most important paper in this regard is *probably* the one due to Kang and Cho [11], referred to as the Locally Linear Reconstruction (LLR) method. The fundamental idea behind the LLR, though simple, is quite intriguing, and it involves a quadratic optimization strategy explained presently. The salient feature of this scheme is that by invoking this optimization, one can systematically determine the model parameters, such as the number of neighbors (k) and the corresponding weights. However, the LLR, as proposed in [11], is computationally intensive. This is where our research comes into the picture: To tackle the computational burden, we propose a strategy of using a Prototype Reduction Scheme (PRS) to quickly and efficiently approximately compute the optimization problem. We formulate this in the paragraph below.

Rationale for the paper: We start with the premise that it is advantageous to compute the above mentioned optimization. However, we seek a strategy by which the associated computational burden can be reduced. Thus, in this paper, we propose a technique[2] for the fast computation of the reconstruction problem, and in particular, for the various classification applications. We advocate that rather than compute the reconstruction for the entire data set, the data be first reduced into a smaller representative subset using a PRS [2], [6], and that the reconstruction (classification) be achieved by invoking the corresponding method on *this* reduced data set. Thus, by completely discarding the points not included by the PRS, we can obtain a reduced set of sample points, using which, in turn, one can solve the quadratic optimization problem. The reader will observe, at once, that this can reduce the computational burden drastically, because the number of points chosen by the PRS is usually a small *fraction* of the total number of points found in the original data set. Our hypothesis, i.e., that the PRS can be effectively used to noticeably reduce the computations and yet yield almost as accurate results, has been verified by testing on benchmark real-life and artificial data tests, as we

[1] Some strategies for speeding up the kNN have been reported in the literature, e.g., in [14].

[2] As a *prima facie* case, to justify the hypothesis of [11], we only consider the two-class problem. The effective definition and computation of the measures for the multi-class problem are open.

shall presently explain. The geometric aspect of this strategy is the following: Although the reconstructed samples are obtained by using the prototypes procured by invoking a PRS, these reconstructed points do not *individually* "optimally" represent their original counterparts. However, *collectively*, they are the best locations for the k-NNs of the points in the training set, which can, in turn, *collectively* represent the points for testing purposes too. This is truly an interesting feature!

2 An Overview : LLR and PRS

Locally Linear Reconstruction: In this section, we briefly explain the LLR [11] for pattern classification and recognition (as considered for instance-based learning), and in particular for the k-NN. The main idea behind LLR originates from the concept of the locally linear embedding (LLE) [16], which is one of widely-used non-linear dimension reduction schemes. Of course, as mentioned earlier, the premise behind NN learning is that if the input vectors are similar, the targets are also similar with a very high likelihood. In order to realize this premise, researchers have used monotonically decreasing kernel functions, with regard to the distance, to assign weights to the neighbors. Along the same vein, in the case of LLR, we attempt to enforce this general premise in the topological space for the k-NN. Indeed, we argue that if it is possible to accurately describe the input vector for a given query by its neighboring reference patterns, it is also possible to predict (estimate) well the target class (value) of the query with a small error. To initiate discussions in this regard, we first state the notation that we shall use (in a d-dimensional feature space), after which we shall formally describe LLR.

- $\underline{X}_i$ is a "query" (i.e., the testing point) in the feature space, and is a $d \times 1$ vector.
- $\underline{\widehat{X}}_i$ is a re-constructed version of $\underline{X}_i$, and is also a $d \times 1$ vector.
- $\boldsymbol{X}^i_{NN}$ is a $d \times k$ matrix, and contains the d-dimensional k-NNs of $\underline{X}_i$.
- $\underline{W}_{i,NN}$ is a $k \times 1$ vector. It is the corresponding weight vector obtained from $\boldsymbol{X}^i_{NN}$. The matrix $\boldsymbol{W}$, which is the collection of $\underline{W}_{i,NN}$'s, is the set of vectors sought for, and $W_{i,j}$ is the set of weights for $\underline{X}_j$ with regard to the sample point $\underline{X}_i$. Observe that $W_{i,j}$ will be zero if $\underline{X}_j$ is not a neighbor of $\underline{X}_i$.
- The matrix $\mathbf{N}$ is the neighborhood indicator matrix whose element $N_{i,j} = 0$ if $\underline{X}_j$ is not a neighbor of $\underline{X}_i$, and is unity otherwise. For ease of notation, $\mathbf{N}(i)$ will represent the NNs of $\underline{X}_i$.

When a query is given, the method first selects the k-nearest neighbors of the query. Once these NN patterns have been selected, the set of weights corresponding to the neighbor are determined by minimizing the LLR error $Err(\boldsymbol{W})$, defined as the sum of the errors E_i as follows: $\sum_i \left\| \underline{X}_i - \underline{W}^T_{i,NN} \underline{X}_j \right\|^2$, where every $\underline{X}_j$ is a NN of $\underline{X}_i$.

The weights, $\boldsymbol{W}$, which minimize the reconstruction error, can be obtained by solving the above minimization problem. Also, since the constraints on the optimization problem differ depending on whether the learning task is a classification or regression problem, the corresponding procedures for solving them are different as well. In particular, for classification tasks, we need to impose two additional constraints on $\boldsymbol{W}$, namely that all the weights must be non-negative, and that the sum of the neighbors' weights must be *unity* for every query. Thus,

$$Err(\boldsymbol{W}) = \frac{1}{2}\sum_i \left\| \underline{X}_i - \underline{W}_{i,NN}^T \boldsymbol{X}_{NN}^{i\ T} \right\|^2 \tag{1}$$
$$= \frac{1}{2}\sum_i \left\{ \underline{X}_i^T \underline{X}_i - 2\underline{X}_i^T \mathbf{X}_{NN}^i \underline{W}_{i,NN} + \underline{W}_{i,NN}^T \boldsymbol{X}_{NN}^{i\ T} \boldsymbol{X}_{NN}^i \underline{W}_{i,NN} \right\}.$$

By examining Eq. (1), we see that we can obtain the weights for the k-NNs of $\underline{X}_i$, $\underline{W}_{i,NN}$, by solving the following optimization problem[3]:

$$Min\ Err(\underline{W}_{i,NN}) = \frac{1}{2}\underline{W}_{i,NN}^T \boldsymbol{X}_{NN}^{i\ T} \boldsymbol{X}_{NN}^i \underline{W}_{i,NN} - \underline{X}_i \boldsymbol{X}_{NN}^i \underline{W}_{i,NN}, \tag{2}$$
$$such\ that\ \ \underline{W}_{i,NN} \geq 0,\ \ \sum_j W_{i,j} = 1\ \ \forall i.$$

After obtaining the weights assigned, we can reconstruct a sample point, $\widehat{\underline{X}}_i$, corresponding to the query $\underline{X}_i$ by a weighted sum of the samples of $\underline{X}_i$'s NNs as follows:

$$\widehat{\underline{X}}_i = \sum_{\underline{X}_j \in \boldsymbol{N}(i)} W_{i,j} \boldsymbol{X}_j. \tag{3}$$

As the reader will observe, although this strategy is expedient, it involves the unavoidable non-trivial computationally intensive optimization. But our position is that it need not be done for all the sample points, but merely for a smaller subset of points which *represent* them - i.e., those obtained by a PRS.

Prototype Reduction Schemes: In non-parametric pattern classification which uses the NN or the $k-$NN rule, each class is described using a set of sample prototypes, and the class of an unknown vector is decided based on the identity of the closest neighbor(s) which are found among all the prototypes. To reduce the number of training vectors, various PRSs have been reported in the literature - two excellent surveys are found in [2], [6]. Rather than embark on yet another survey of the field, we mention here a *few* representative methods of the "zillions" that have been reported. One of the first of its kind is the Condensed Nearest Neighbor (CNN) rule [10]. The reduced set produced by the CNN, however, customarily includes "interior" samples, which can be completely eliminated, without altering the performance of the resultant classifier. Accordingly, other methods have been proposed successively, such as the Reduced Nearest Neighbor (RNN) rule, the Prototypes for Nearest Neighbor (PNN) classifiers [5], the Selective Nearest Neighbor (SNN) rule [15], two modifications of the CNN [18], the Edited Nearest Neighbor (ENN) rule [7], and the non-parametric data reduction method [9]. Besides these, the Vector Quantization (VQ) and the Bootstrap techniques have also been reported as being extremely effective approaches to data reduction. Recently, Support Vector Machines (SVM) [4] have proven to possess the capability of extracting

[3] The quadratic programming problem, $\min \frac{1}{2}\underline{U}^T \boldsymbol{H}\underline{U} + \underline{B}^T\underline{U}$, such that $\boldsymbol{A}\underline{U} \leq 0$, $\boldsymbol{A}_{eq}\underline{U} = \underline{b}_{eq}$, and $\underline{l}_b \leq \underline{U} \leq \underline{u}_b$, (where $\boldsymbol{H}$, $\boldsymbol{A}$, and $\boldsymbol{A}_{eq}$ are matrices, and $\underline{B}$, $\underline{b}_{eq}$, $\underline{l}_b$, $\underline{u}_b$, and $\underline{U}$ are vectors) defines a set of lower and upper bounds on the design variables, $\underline{U}$, so that the solution is in the range $\underline{l}_b \leq \underline{U} \leq \underline{u}_b$.

vectors that support the boundary between any two classes. Thus, they have been used satisfactorily to represent the global distribution structure.

In selecting prototypes, vectors near the boundaries between the classes have to be considered to be more significant, and the created prototypes need to be adjusted towards the classification boundaries so as to yield a higher performance. Based on this philosophy, Kim and Oommen [12], [13] proposed a new hybrid approach (HYB) that involved two distinct phases, namely, those of selecting and adjusting [12]. To overcome the computational burden for "large" datasets, they also proposed a recursive HYB in [13]. In [13], the data set is sub-divided recursively into smaller subsets to filter out the "useless" internal points. Subsequently, a conventional PRS (i.e., HYB) processes the smaller subsets of data points that effectively sample the entire space to yield *subsets* of prototypes – one set of prototypes for each subset. The prototypes, which result from each subset, are then coalesced, and processed again by the PRS to yield more refined prototypes. In this manner, prototypes which are in the interior of the Voronoi boundaries, and are thus ineffective in the classification, are eliminated at the subsequent invocations of the PRS, *noticeably* reducing the PRS's processing time.

This overview of the state-of-the-art of PRSs should be sufficient to help us proceed in formulating our solution to the problem at hand.

3 Schema for the Proposed Solution

Our goal is to "quickly" find out the class of a query point in the input feature space after reconstructing an approximated version of the corresponding sample using its NNs. However, rather than reconstruct the approximated data sample using the entire training set, we advocate that the data be first reduced into a smaller representative subset using a PRS, and that the data point be estimated by invoking a reconstruction scheme on *this* reduced data set. Thereafter, the classification accuracy of the $k-$NN classifier is compared. Thus, the proposed scheme can be formalized as follows:

Algorithm 1. PRS_LLR

Input: The original Training Set, T.

Output: Testing by utilizing a fast reconstruction of the approximated query point using a reduced data set rather than the entire training set.

Assumption 1: The algorithm has access to a PRS such as the CNN, PNN or HYB.

Assumption 2: The algorithm has access to the LLR algorithm mentioned previously.

Method:

Step 1: Select the representative set, Y, from the training set T by resorting to a PRS.

Step 2: Find the closest neighbors, $\boldsymbol{X}_{NN}^{i}$, for a query $\underline{X}_i$ from Y, rather than from T.

Step 3: Compute corresponding weight vector, $\underline{W}_{i,NN}$, using LLR and a k_1-NN rule.

Step 4: Reconstruct $\widehat{\underline{X}}_i$ with LLR using $\boldsymbol{X}_{NN}^{i}$ and $\underline{W}_{i,NN}$, and the k_1-NN rule.

Step 5: Classify $\widehat{\underline{X}}_i$ by comparing it with the elements of Y using the best k_2-NN rule.

End Algorithm PRS_LLR

We would like to emphasize that there are a few fundamental differences between what we propose and the original LLR method proposed in [11]. First of all, we observe that the computation of the LLR weights does not involve the entire training set T, but

a representative set, Y, derived from it using a PRS. Secondly, we note that the weights that are computed for the LLR involve a NN rule, using k_1 neighbors, where the latter is the pre-determined degree of the NN classifier used for the training phase. But once the reconstructed point is obtained, we now have the freedom of testing it using the most suitable NN classifier, which may not necessarily be a k_1-NN classifier. Indeed, as in any PR problem, given a training set, the practitioner has the freedom to choose the best NN classifier that suits his application. In the same vein, in our case, we choose the best "Testing" NN classifier (a k_2-NN classifier) for the application domain, using the modified "Training" set, Y, and the modified testing sample, $\underline{\widehat{X}}_i$. It turns out that usually, k_2 is quite distinct from k_1!

We shall now demonstrate the power of **Algorithm PRS_LLR**.

4 Experimental Set-Up, Results and Evaluation

Experimental Data: The proposed scheme has been tested and compared with the conventional LLR method reported in the literature. This was done by performing experiments on both "artificial" and "real-life" data sets[4]. In each case, the sample vectors of each data set was divided into two subsets of equal size $T1$ and $T2$ (typically, used for training and validation, alternatively). The computation was done on each subset and subsequently averaged.

In our experiments, the four artificial data sets "Non_normal 2, 3" and "Non_linear 2, 3", were generated with different sizes for the testing and training sets, and had cardinalities of 500 and $5,000$ respectively. The data sets "Ionosphere", "Sonar", and "Arrhythmia", which are real benchmark data sets, are cited from the UCI Machine Learning Repository [3].

The data set named "Non_normal" (in short, "Non_n"), which has also been employed as a benchmark experimental data set for numerous experimental set-ups, was generated from a mixture of four 8-dimensional Gaussian distributions as desribed in detail in [8]). The data set named "Non_linear" (in short, "Non_l") which has a strong non-linearity at its boundary, was generated artificially from a mixture of four normal variables as described in [13].

Experimental Parameters: Choosing the parameters of PRSs play an important role in determining the quality of the solution. The parameters[5] for the PRSs[6] were: Since the number of prototypes depends on the characteristics of the data set, the number of

[4] More extensive results for other data sets are available, but omitted here in the interest of space.

[5] These parameters are included here for the sake of researchers who would like to duplicate the results.

[6] The reader should observe that, as mentioned previously, any PRS can be employed to obtain the reduced set, Y. In the present paper, only three methods, namely CNN, PNN, and HYB have been used in the testing. The main reason for choosing these is as follow: First of all, the prototype vectors obtained with CNN and PNN are *selected* and *created*, respectively. On the other hand, for HYB, the prototypes are initially *selected*, after which they are adjusted. Finally, for all the methods, the final number of prototypes is not a quantity that is controlled or determined automatically.

iterations is predetermined by the size of T. Hence, CNN and PNN had no parameters. In HYB, we invoked a hybridized version of the SVM and an LVQ3-type algorithm, both of which are available in publicly distributed packages. The SVM was employed to determine the initial code book vectors for the LVQ3. The parameters for the LVQ3 learning are specified in [12]. For instance, the parameters for the data set "Adult4" were $\alpha = 0.05$, $\epsilon = 0.06$, $w = 0.35$, $\eta = 5{,}600$.

Selecting Prototype Vectors: In order to evaluate the proposed classification mechanisms, we first selected the prototype vectors from the experimental data sets using the CNN, PNN, and HYB algorithms. In HYB, we selected initial prototypes using a SVM algorithm. After this selection, we invoked a phase in which the optimal positions (i.e., with regard to classification) were learned with an LVQ3-type scheme. For the SVM and LVQ3 programs, we utilized publicly-available software packages. For example, we can see that the numbers of selected prototype vectors of the "Non_n2" dataset with CNN are $(64, 66)$, $(56, 380)$, and $(63, 57)$, respectively. Each of them is considerably smaller than the size of the original data set, $(500, 500)$. Using the selected vectors as a representative of the training data set, we can significantly reduce the cardinality of the dataset (and the consequential computations) without noticeably degrading the performance. The reduction of the classification processing time follows as a natural consequence. As an observation, we also mention that the reduction rate increased dramatically as the size of the data sets was increased.

Experimental Results: To illustrate the method, consider Figure 1 which shows the plots of the 2-dimensional data set $\left\{(x_2^i, x_4^i)^T\right\}_{i=1}^{50}$ projected from the original four dimensional "Iris2" data set. The figure on the left is the original data set, where the points of two classes are represented by '$\times$' and '$+$', respectively. The figure on the right is the reconstructed data set from the prototypes, rather than the entire samples using LLR. Here, the prototypes are extracted by CNN and represented by '$\otimes$' and '$\oplus$', respectively. The reader should observe the non-intuitive properties of the scheme by studying Figure 1. Although the samples shown in the figure on the right (given by '$\times$' and '$+$' respectively), are reconstructed by using the prototypes '$\otimes$' and '$\oplus$', respectively from the figure on the left, the reconstructed points do not *individually* "optimally" represent their original counterparts. However, *collectively*, they are the best locations for the k_1-NNs which can, in turn, *collectively* represent the points.

Tables 1 and 2 show the run-time characteristics of the proposed scheme for the artificial data sets and the other benchmark data sets. With regard to notation, in these tables, the abbreviations WHL, CNN, PNN, and HYB correspond to the experimental methods employed for the WHoLe data set, and the prototypes extracted with the CNN, PNN, and HYB methods, respectively. Analogously, in the case of WHL, the data complexities (classification accuracies) and the corresponding processing CPU-times were measured for the whole data set, and for CNN, PNN, and HYB, the measures were computed for the corresponding extracted prototypes.

By examining the results, it is clear that the classification accuracies, for the benchmark databases can be measured quite efficiently and fairly accurately by first invoking the corresponding PRS techniques. To clarify this, consider, for example, the accuracies obtained for the samples reconstructed with NN = 5 for "Non_n2". The classification

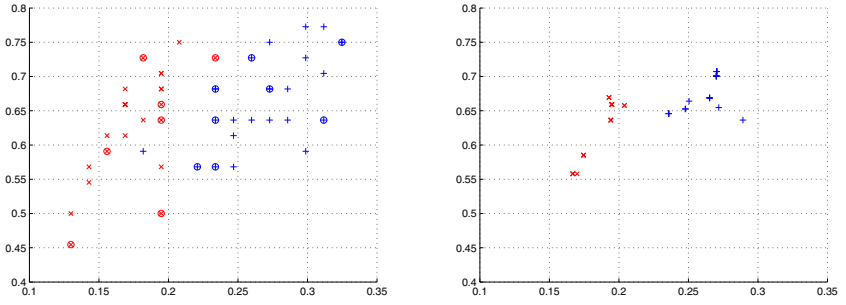


Fig. 1. Plots of the 2-dimensional vectors $\left\{ \binom{x_2^i}{x_4^i} \right\}_{i=1}^{50}$ generated from the original 4-dimensional "Iris2" data set. The details of the picture are discussed in the text.

Table 1. A comparison of classification accuracies (%) for the samples locally reconstructed with the experimental data sets and their prototypes extracted with CNN, PNN, and HYB, where each evaluation sample was reconstructed with the k_1 nearest neighbors of cardinalities *1, 3, 5, 7, 9, 11, 13*. The number in parenthesis in each entry represents the "order" k_2, of the corresponding "Testing" classifier, using which the respective accuracy was obtained.

Datasets	PRS	NN: k_1=1	NN: k_1=3	NN: k_1=5	NN: k_1=7	NN: k_1=9	NN: k_1=11	NN: k_1=13
Non_n3	WHL	94.50 (11)	94.52 (13)	94.54 (13)	94.54 (13)	94.54 (13)	94.50 (13)	94.52 (13)
	CNN	94.50 (11)	94.62 (11)	94.56 (13)	94.57 (13)	94.58 (13)	94.55 (13)	94.54 (13)
	PNN	94.40 (7)	94.53 (13)	94.56 (13)	94.56 (13)	94.53 (13)	94.60 (13)	94.50 (13)
	HYB	71.70 (9)	42.27 (1)	42.30 (1)	42.29 (1)	42.27 (1)	42.24 (1)	42.34 (1)
Non_l3	WHL	91.08 (11)	91.16 (13)	91.07 (13)	91.14 (13)	91.12 (13)	91.12 (13)	91.20 (11)
	CNN	90.46 (9)	90.06 (7)	89.96 (9)	89.83 (9)	89.74 (9)	89.71 (9)	89.70 (9)
	PNN	87.83 (9)	88.67 (7)	88.86 (7)	89.06 (7)	89.10 (7)	89.09 (7)	89.12 (7)
	HYB	88.04 (13)	88.36 (13)	88.18 (13)	88.18 (13)	88.12 (13)	88.20 (13)	88.24 (13)
Ionos	WHL	78.69 (1)	77.55 (1)	76.98 (1)	76.13 (1)	76.42 (1)	75.85 (1)	75.85 (1)
	CNN	81.81 (1)	80.39 (1)	77.84 (1)	77.27 (1)	75.85 (1)	74.43 (1)	74.14 (1)
	PNN	82.67 (1)	83.52 (3)	82.95 (3)	83.23 (3)	82.38 (3)	82.38 (3)	81.81 (3)
	HYB	83.23 (1)	80.96 (1)	77.27 (1)	78.12 (3)	78.97 (3)	78.40 (3)	78.69 (3)
Sonar	WHL	82.21 (1)	83.65 (3)	84.61 (3)	84.13 (3)	84.13 (3)	83.65 (3)	83.65 (3)
	CNN	79.80 (1)	79.80 (1)	78.36 (1)	79.32 (1)	78.36 (1)	79.32 (1)	79.80 (1)
	PNN	82.69 (1)	82.21 (1)	81.25 (1)	81.73 (1)	81.25 (1)	79.80 (1)	79.80 (1)
	HYB	80.76 (1)	79.80 (1)	79.80 (1)	79.32 (1)	79.32 (1)	78.36 (1)	78.84 (1)
Arrhy	WHL	97.56 (1)	97.56 (1)	97.56 (1)	97.78 (1)	97.78 (1)	97.34 (1)	97.34 (1)
	CNN	96.46 (1)	96.46 (1)	96.46 (1)	96.01 (1)	95.79 (1)	94.91 (1)	95.13 (1)
	PNN	99.11 (1)	99.11 (1)	98.89 (1)	98.67 (1)	–	–	–
	HYB	99.11 (1)	98.89 (1)	98.67 (1)	98.45 (1)	98.45 (1)	98.89 (1)	99.11 (1)

accuracies of WHL, CNN, PNN, and HYB are 94.50, 94.50, 94.60, and 71.90 (%), respectively, where the quantities mentioned in parenthesis in each row represent the classification accuracies that are obtained with the *11*-NN, *9*-NN, *7*-NN, and *5*-NN classifiers, respectively. But with regard to computation, the processing CPU-times of these

Table 2. A comparison of the processing CPU-times (seconds) required for the samples locally reconstructed with the experimental data sets and their prototypes. Here, the prototypes were extracted with CNN, PNN, and HYB, respectively. Thereafter, each evaluation sample was reconstructed with the nearest neighbors of cardinalities *1, 3, 5, 7, 9, 11, 13*.

Datasets	PRS	NN: k_1=1	NN: k_1=3	NN: k_1=5	NN: k_1=7	NN: k_1=9	NN: k_1=11	NN: k_1=13
Non_n3	WHL	91.22	112.30	117.02	128.06	148.27	152.55	167.41
	CNN	80.38	76.70	75.30	83.92	94.55	99.50	111.22
	PNN	114.22	139.23	135.19	152.86	183.89	174.75	192.83
	HYB	81.32	92.41	100.21	111.61	120.64	125.44	138.72
Non_l3	WHL	121.36	130.36	141.44	151.47	185.13	208.06	227.08
	CNN	58.42	65.59	71.67	82.78	98.50	108.94	125.95
	PNN	80.56	88.03	95.75	110.23	125.33	138.28	136.41
	HYB	90.53	97.38	105.88	126.09	151.14	167.52	188.38
Ionos	WHL	2.92	6.88	6.55	6.63	7.78	7.21	7.52
	CNN	2.85	6.72	6.60	6.57	6.94	6.96	7.29
	PNN	2.79	6.79	6.33	6.80	6.80	7.07	7.24
	HYB	2.87	6.85	6.57	6.66	6.82	7.07	7.13
Sonar	WHL	1.86	3.56	3.85	3.90	4.01	4.06	4.24
	CNN	2.00	3.46	3.90	3.87	4.01	4.18	4.21
	PNN	1.92	3.42	3.85	4.04	4.17	4.13	4.32
	HYB	1.90	3.60	3.92	3.95	4.13	4.21	4.34
Arrhy	WHL	98.94	107.27	109.28	114.02	119.34	123.80	130.98
	CNN	67.86	74.30	77.22	79.20	83.77	87.66	95.20
	PNN	64.36	70.67	73.05	76.17	–	–	–
	HYB	71.88	85.20	83.27	90.88	92.61	96.69	101.89

methods are 22.28, 20.28, 20.73, and 19.56 seconds, respectively[7] – which represents an advantage of about 12%. The effect is more marked in the case of large data sets. For example, in the case of the "Non_l3" data set, the accuracy measures of WHL, CNN, PNN, and HYB are 91.07, 89.96, 88.86, and 88.18 (%), respectively, while the processing times involved by using the PRSs are much smaller – namely 71.67, 95.75, and 105.88 seconds respectively, instead of 141.44 seconds required for the entire data set. Similar observations can also be made for the other benchmark data sets. But, in general, as an overall conclusion we believe that we can assert that a PRS can be effectively invoked to optimize the Locally Linear Reconstruction process for PR applications.

5 Conclusions

In this paper, we have considered how we can use the principles of Prototype Reduction Schemes (PRSs) to optimize the computations involved in the well-known families of k-Nearest Neighbor (k-NN) rules. Although k-NN rules have been extensively studied, recently, an implementation of the k-NN, named as the Locally Linear Reconstruction

[7] The times recorded are the times required for the MATLAB computation on a PC with a CPU speed of 2.40GHz and RAM 2GB, and operating on a Windows platform.

(LLR) [11], which invokes a quadratic optimization process has been proposed. The latter method is capable of systematically setting model parameters, such as the number of neighbors (k) and the weights. Our aim, in this paper, was to optimize the computation time required for LLR by using a PRS. We have proposed a strategy of using a PRS to efficiently compute the optimization problem. We have demonstrated that by completely discarding the points not included by the PRS, we can obtain a reduced set of sample points, using which, in turn, the quadratic optimization problem can be computed. The accuracies of proposed method is comparable to those obtained with the original training set (i.e., the one which considers all the data points) even though the computations required are noticeably less (the proposed method sometimes requiring only about 50% of the time). The proposed method has been tested on artificial and real-life data sets, and the results obtained are quite promising, and could have potential in PR applications. An avenue for further research involves developing alternate stochastic learning methods by which the query sample can be estimated accurately and quickly.

References

1. Atkeson, C.G., Moore, A.W., Schaal, S.: Locally weighted learning. Artifical Intelligence Review 11(5), 11–73 (1997)
2. Bezdek, J.C., Kuncheva, L.I.: Nearest prototype classifier designs: An experimental study. International Journal of Intelligent Systems 16(12), 1445–11473 (2001)
3. Blake, C.L., Merz, C.J.: UCL Machine Learning Databases. University of California, Department of Information and Computer Science, Irvine, CA, Can also be downloaded from http://www.ics.uci.edu/mlearn/MLRepository.html
4. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2(2), 121–167 (1998)
5. Chang, C.L.: Finding prototypes for nearest neighbor classifiers. IEEE Trans. Computers 23(11), 1179–1184 (1974)
6. Dasarathy, B.V.: Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos (1991)
7. Devijver, P.A., Kittler, J.: On the edited nearest neighbor rule. In: Proc. 5th Int. Conf. on Pattern Recognition, Miami, Florida, pp. 72–80 (1980)
8. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, San Diego (1990)
9. Fukunaga, K., Mantock, J.M.: Nonparametric data reduction. IEEE Trans. Pattern Anal. and Machine Intell. 6(1), 115–118 (1984)
10. Hart, P.E.: The condensed nearest neighbor rule. IEEE Trans. Inform. Theory 14, 515–516 (1968)
11. Kang, P., Cho, S.: Locally linear reconstruction for instance-based learning. Pattern Recognition 41, 3507–3518 (2008)
12. Kim, S.-W., Oommen, B.J.: Enhancing prototype reduction schemes with LVQ3-type algorithms. Pattern Recognition 36(5), 1083–1093 (2003)
13. Kim, S.-W., Oommen, B.J.: Enhancing prototype reduction schemes with recursion: A method applicable for “large” data sets. IEEE Trans. Systems, Man, and Cybernetics - Part B 34(3), 1384–1397 (2004)

14. Liu, T., Moore, A., Gray, A.: Efficient exact k-NN and nonparametric classification in high dimensions. In: Proc. of Neural Information Processing Systems (2003)
15. Ritter, G.L., Woodruff, H.B., Lowry, S.R., Isenhour, T.L.: An algorithm for a selective nearest neighbor rule. IEEE Trans. Inform. Theory 21, 665–669 (1975)
16. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500), 2323–2326 (2000)
17. Roweis, S.T., Saul, L.K.: Think globally, fit locally: unsupervised learning of nonlinear manifolds. Journal of Machine Learning Research 4, 119–155 (2003)
18. Tomek, I.: Two modifcations of CNN. IEEE Trans. Syst., Man and Cybern. 6(6), 769–772 (1976)

Efficient Two Stage Voting Architecture for Pairwise Multi-label Classification

Gjorgji Madjarov, Dejan Gjorgjevikj, and Tomche Delev

Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University, Rugjer Boshkovikj bb, 1000 Skopje, R. of Macedonia
{madzarovg,dejan,tdelev}@feit.ukim.edu.mk

Abstract. A common approach for solving multi-label classification problems using problem-transformation methods and dichotomizing classifiers is the pair-wise decomposition strategy. One of the problems with this approach is the need for querying a quadratic number of binary classifiers for making a prediction that can be quite time consuming especially in classification problems with large number of labels. To tackle this problem we propose a two stage voting architecture (TSVA) for efficient pair-wise multiclass voting to the multi-label setting, which is closely related to the calibrated label ranking method. Four different real-world datasets (enron, yeast, scene and emotions) were used to evaluate the performance of the TSVA. The performance of this architecture was compared with the calibrated label ranking method with majority voting strategy and the quick weighted voting algorithm (QWeighted) for pair-wise multi-label classification. The results from the experiments suggest that the TSVA significantly outperforms the concurrent algorithms in term of testing speed while keeping comparable or offering better prediction performance.

Keywords: Multi-label, classification, calibration, ranking.

1 Introduction

Traditional single-label classification is concerned with learning from set of examples that are associated with a single label λ_i from a finite set of disjoint labels $L = \{\lambda_1, \lambda_2, ..., \lambda_Q\}$, $Q > 1$. If $Q = 2$, then the learning problem is called a binary classification problem, while if $Q > 2$, then it is called a multi-class classification problem. On the other hand, multi-label classification is concerned with learning from a set of examples $S = \{(x_1, Y_1), (x_2, Y_2), ..., (x_p, Y_p)\}$ ($x_i \in X$, X denote the domain of examples) where each of the examples is associated with a set of labels $Y_i \subseteq L$.

Many classifiers were originally developed for solving binary decision problems, and their extensions to multi-class and multi-label problems are not straight-forward. Because of that, a common approach to address the multi-label classification problem is utilizing class binarization methods, i.e. decomposition of the problem into several binary subproblems that can then be solved using

J. Li (Ed.): AI 2010, LNAI 6464, pp. 164–173, 2010.

a binary base learner. The simplest strategy in the multi-label setting is the one-against-all strategy also referred to as the binary relevance method. It addresses the multi-label classification problem by learning one classifier (model) M_k $(1 \leq k \leq Q)$ for each class, using all the examples labeled with that class as positive examples and all other (remaining) examples as negative examples. At query time, each binary classifier predicts whether its class is relevant for the query example or not, resulting in a set of relevant labels.

Another approach for solving the multi-label classification problem using binary classisifers is pair-wise classification or round robin classification [1][2]. Its basic idea is to use $Q * (Q-1)/2$ classifiers covering all pairs of labels. Each classifier is trained using the samples of the first label as positive examples and the samples of the second label as negative examples. To combine these classifiers, the pair-wise classification method naturally adopts the majority voting algorithm. Given a test instance, each classifier delivers a prediction for one of the two labels. This prediction is decoded into a vote for one of the labels. After the evaluation of all $Q * (Q-1)/2$ classifiers the labels are ordered according to their sum of votes. To predict only the relevant classes for each instance a label ranking algorithm is used. Label ranking studies the problem of learning a mapping from set of instances to rankings over a finite number of predefined labels. It can be considered as a natural generalization of conventional classification, where only a single label (the top-label) is requested instead of a ranking of all labels.

Brinker et al. [3] propose a conceptually new technique for extending the common pair-wise learning approach to the multi-label scenario named calibrated label ranking. The key idea of calibrated label ranking is to introduce an artificial (calibration) label λ_0, which represents the split-point between relevant and irrelevant labels. The calibration label λ_0 is assumed to be preferred over all irrelevant labels, but all relevant labels are preferred over it. At prediction time (when majority voting strategy is usually used), one will get a ranking over $Q+1$ labels (the Q original labels plus the calibration label). The calibrated label ranking is considered a combination of both multi-label classification and ranking.

Besides the majority voting that is usually used strategy in the prediction phase of the calibrated label ranking algorithm, Park et al. [4] propose another more effective voting algorithm named Quick Weighted Voting algorithm (QWeighted). QWeighted computes the class with the highest accumulated voting mass avoiding the evaluation of all possible pair-wise classifiers. It exploits the fact that during a voting procedure some classes can be excluded from the set of possible top rank classes early in the process when it becomes clear that even if they reach the maximal voting mass in the remaining evaluations they can no longer exceed the current maximum. Pair-wise classifiers are selected depending on a voting loss value, which is the number of votes that a class has not received. The voting loss starts with a value of zero and increases monotonically with the number of performed preference evaluations. The class with the current minimal loss is the top candidate for the top rank class. If all

preferences involving this class have been evaluated (and it still has the lowest loss), it can be concluded that no other class can achieve a better ranking. Thus, the QWeighted algorithm always focuses on classes with low voting loss. An adaptation of QWeighted to multi-label classification (QWeightedML) [5] is to repeat the process while all relevant labels are not determined i.e. until the returned class is the artificial label λ_0, which means that all remaining classes will be considered to be irrelevant.

In this paper we propose an efficient Two Stage Voting Architecture (TSVA) that modifies the majority voting algorithm for calibrated label ranking technique [6]. We have evaluated the performance of this architecture on a selection of multi-label datasets that vary in terms of problem domain and number of labels. The results demonstrate that our modification outperforms the majority voting algorithm for pair-wise multi-label classification and the QWeightedML [5] algorithm in terms of testing speed, while keeping comparable prediction results.

For the readers' convenience, in Section 2 we will briefly introduce notations and evaluation metrics used in multi-label learning. The Two Stage Voting Architecture is explained in Section 3. The experimental results that compare the performance of the proposed TSVA with concurrent methods are presented in Section 4. Section 5 gives a conclusion.

2 Preliminaries

Let X denote the domain of instances and let $L = \{\lambda_1, \lambda_2, ..., \lambda_Q\}$ be the finite set of labels. Given a training set $S = \{(x_1, Y_1), (x_2, Y_2), ..., (x_p, Y_p)\}$ $(x_i \in X, Y_i \subseteq L)$, the goal of the learning system is to output a multi-label classifier $h : X \rightarrow 2^L$ which optimizes some specific evaluation metric. In most cases however, instead of outputting a multi-label classifier, the learning system will produce a real-valued function of the form $f : X \times L \rightarrow R$. It is supposed that, given an instance x_i and its associated label set Y_i, a successful learning system will tend to output larger values for labels in Y_i than those not in Y_i, i.e. $f(x_i, y_1) > f(x_i, y_2)$ for any $y_1 \in Y_i$ and $y_2 \notin Y_i$. The real-valued function $f(\bullet, \bullet)$ can be transformed to a ranking function $rank_f(\bullet, \bullet)$, which maps the outputs of $f(x_i, y)$ for any $y \in L$ to $\{\lambda_1, \lambda_2, ..., \lambda_Q\}$ such that if $f(x_i, y_1) > f(x_i, y_2)$ then $rank_f(x_i, y_1) < rank_f(x_i, y_2)$. Note that the corresponding multi-label classifier $h(\bullet)$ can also be derived from the function $f(\bullet, \bullet) : h(x_i) = \{y | f(x_i, y) > t(x_i); y \in L\}$, where $t(\bullet)$ is a threshold function which is usually set to be the zero constant function. Performance evaluation of multi-label learning system is different from that of classical single-label learning system. The following multi-label evaluation metrics proposed in [7] are used in this paper:

(1) Hamming loss: evaluates how many times an instance-label pair is misclassified, i.e. a label not belonging to the instance is predicted or a label belonging to the instance is not predicted. The performance is perfect when $hloss_S(h) = 0$. The smaller the value of $hloss_S(h)$, the better the performance. This metric is given by

$$hloss_S(h) = \frac{1}{p}\sum_{i=1}^{p}\frac{1}{Q}\left|h(x_i)\Delta Y_i\right| \tag{1}$$

where Δ stands for the symmetric difference between two sets and Q is the total number of possible class labels. Note that when $|Y_i| = 1$ for all instances, a multi-label system reduces to multi-class single-label one and the hamming loss becomes $2/Q$ times the usual classification error.

While hamming loss is based on the multi-label classifier $h(\bullet)$, the other four metrics are defined based on the real-valued function $f(\bullet,\bullet)$ that takes into account the ranking quality of different labels for each instance:

(2) One-error: evaluates how many times the top-ranked label is not in the set of proper labels of the instance. The performance is perfect when $one - error_S(f) = 0$. The smaller the value of $one - error_S(f)$, the better the performance. This evaluation metric is given by:

$$one - error_S(f) = \frac{1}{p}\sum_{i=1}^{p}\left[\!\left[\left[\arg\max_{y\in Y} f(x_i,y)\right] \notin Y_i\right]\!\right] \tag{2}$$

where for any predicate π, $[\![\pi]\!]$ equals 1 if π holds and 0 otherwise. Note that, for single-label classification problems, the one-error is identical to ordinary classification error.

(3) Coverage: evaluates how far, on the average we need to go down the list of ranked labels in order to cover all the proper labels of the instance. The smaller the value of $coverage_S(f)$, the better the performance.

$$coverage_S(f) = \frac{1}{p}\sum_{i=1}^{p}\max_{y\in Y_i} rank_f(x_i,y) - 1 \tag{3}$$

(4) Ranking loss: evaluates the average fraction of label pairs that are reversely ordered for the particular instance given by:

$$rloss_S(f) = \frac{1}{p}\sum_{i=1}^{p}\frac{|D_i|}{|Y_i|\left|\bar{Y}_i\right|} \tag{4}$$

where $D_i = \left\{f(y_1,y_2)|f(x_i,y_1) \le f(x_i,y_2),(y_1,y_2)\in Y_i\times\bar{Y}_i\right\}$, while $\bar{Y}$ denotes the complementary set of Y in L. The smaller the value of $rloss_S(f)$, the better the performance, so the performance is perfect when $rloss_S(f) = 0$.

(5) average precision: evaluates the average fraction of labels ranked above a particular label $y \in Y$ that actually are in Y. The performance is perfect when $avgprec_S(f) = 1$; the bigger the value of $avgprec_S(f)$, the better the performance. This metric is given by:

$$avgprec_S(f) = \frac{1}{p}\sum_{i=1}^{p}\frac{1}{|Y_i|}\sum_{y\in Y_i}\frac{|L_i|}{rank_f(x_i,y)} \tag{5}$$

where $L_i = \{y'|rank_f(x_i,y') \le rank_f(x_i,y), y'\in Y_i\}$.

Note that in the rest of this paper, the performances of the multi-label learning algorithms are evaluated based on the five metrics explained above.

3 Two Stage Voting Architecture (TSVA)

Conventional pair-wise approach learns a model M_{ij} for all combinations of labels λ_i and λ_j with $1 \leq i < j \leq Q$. This way $Q * (Q-1)/2$ different pair-wise models are learned. Each pearwise model M_{ij} is learned with the examples labelled with label λ_i as positive examples and the examples labelled with λ_j as negative examples. The main disadvantage of this approach is that in the prediction phase a quadratic number of base classifiers (models) have to be consulted for each test example.

Further, as a result of introducing the artificial calibration label λ_0 in the calibrated label ranking algorithm, the number of the base classifiers is increased by Q i.e. additional set of Q binary preference models M_{0k} $(1 \leq k \leq Q)$ is learned. The models M_{0k} that are learned by a pair-wise approach to calibrated ranking, and the models M_k that are learned by conventional binary relevance are equivalent. At prediction time (when standard majority voting algorithm is usually used) each test instance needs to consult all the models (classifiers) in order to rank the labels by their order of preference. This results in slower testing, especially when the number of the labels in the problem is big.

In this paper we propose an efficient two stage voting architecture which modifies the majority voting algorithm for the calibrated label ranking technique. It reduces the number of base classifiers that are needed to be consulted in order to make a final prediction for a given test instance. The number of base classifiers that are trained by the calibrated label ranking algorithm and the TSVA in the learning process is equivalent.

The proposed (TSV) architecture is organized in two layers. In the first layer of the architecture Q classifiers are located, while in the second layer of the architecture the rest $Q * (Q-1)/2$ classifiers are located. All of the classifiers in the first layer are the binary relevance models M_{0k}, while in the second layer of the architecture the pair-wise models M_{ij} are located. Each model M_{0k} from the first layer is connected with $Q-1$ models M_{ij} from the second layer, where $k = i$ or $k = j$ $(1 \leq i \leq Q-1, i+1 \leq j \leq Q)$. An example of TSVA for solving four-class multi-label classification problems is shown on Fig. 1.

At prediction time, each model M_{0k} of the first layer of the architecture tries to determine the relevant labels for the corresponding test example. Each model M_{0k} gives the probability (the output value of model M_{0k} is convert to probability) that the test example is associated with the label λ_k. If that probability is appropriately small (under some threshold), we can conclude that the artificial calibration label λ_0 is preferred over the label λ_k i.e. the label λ_k belongs to the set of irrelevant labels. In such case, one can conclude that for the test example, the pair-wise models of the second layer M_{ij} where $i = k$ or $j = k$, need not be consulted, because the binary relevance model M_{0k} from the first layer has already made a decision that the label λ_k belongs to the set of irrelevant labels.

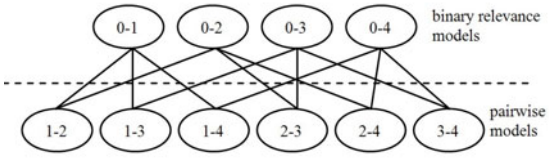


Fig. 1. TSV Architecture

For each test example for which it is known that the label λ_k belongs to the set of irrelevant labels, the number of models that should be consulted decreases for $Q - 1$.

In order to make a decision which labels belong to the set of irrelevant labels i.e. which pair-wise models M_{ij} from the second layer do not have to be consulted a threshold t $(0 \leq t \leq 1)$ is introduced.

According to the previously mentioned, in TSVA every test instance first consults all binary relevance models M_{0k} of the first layer of the architecture. If the corresponding model M_{0k} $(1 \leq k \leq Q)$ response with a probability that is above the threshold t, the test instance is then forwarded only to the models M_{ij} of the second layer of the architecture that are associated to the model M_{0k}. The pair-wise model M_{ij} from the second layer is connected to the binary relevance models M_{0i} and M_{0j}. This does not mean that the model M_{ij} has to be consulted twice, if the prediction probabilities of the models M_{0i} and M_{0j} are both above the threshold t. Instead the model M_{ij} is consulted only once and its prediction is decoded into a vote for one of the labels λ_i or λ_j. If the prediction of one of the models M_{0i} and M_{0j} results with probability under the threshold t, the corresponding model M_{ij} is not consulted and the vote from this model goes to the label which binary relevance model prediction probability is above the threshold t.

By increasing the value of the threshold, the number of consulted pair-wise models decreases. If $t = 1$ the test instance is not forwarded to the second layer of the architecture and the TSVA becomes binary relevance method. On the other hand, if $t = 0$, all pair-wise models of the second layer are consulted and the TSVA becomes calibrated label ranking method with majority voting.

4 Experimental Results

In this section, we present the results of our experiments with several multi-label classification problems. The performance was measured on the problem of recognition of text, music, image and gene function.

Here, the performance of the TSV architecture is compared with the calibrated label ranking method with majority voting strategy for pair-wise multi-label classification (CLR-S) and the QWeightedML algorithm [5].

The training and testing of the TSVA was performed using a custom developed application that uses the MULAN library [8] for the machine learning framework Weka [9]. The LIBSVM library [10] utilizing the SVMs with radial basis kernel

Table 1. Datasets

	scene	yeast	enron	emotions
Domain	image	biology	text	music
Training Instances	1211	1500	1123	391
Test Instances	1159	917	579	202
Features	294	103	1001	72
Labels	6	14	53	6

Table 2. The evaluation of each method for every dataset

	t	Evaluation Metric	CLR-S	QWeightedML	TSVA
enron	0.03	**Hamming Loss**	0.0476	0.0481	0.0501
		One-error	0.2297	0.2262	0.2193
		Coverage	11.5198	20.3333	14.4317
		Ranking Loss	0.0756	0.1516	0.0969
		Avg. Precision	0.7018	0.6543	0.6970
		Testing time (s)	605.06	174.31	147.57
emotions	0.25	**Hamming Loss**	0.2566	0.2623	0.2590
		One-error	0.3812	0.3762	0.3663
		Coverage	2.4059	2.8465	2.3960
		Ranking Loss	0.2646	0.3381	0.2612
		Avg. Precision	0.7215	0.6795	0.7242
		Testing time (s)	2.56	1.67	1.34
yeast	0.15	**Hamming Loss**	0.1903	0.1909	0.1906
		One-error	0.2334	0.2301	0.2300
		Coverage	6.2758	8.6215	6.7633
		Ranking Loss	0.1632	0.2934	0.1805
		Avg. Precision	0.7685	0.7003	0.7641
		Testing time (s)	104.34	60.39	54.65
scene	0.02	**Hamming Loss**	0.0963	0.0956	0.0946
		One-error	0.2349	0.2349	0.2366
		Coverage	0.4883	0.7073	0.4974
		Ranking Loss	0.0779	0.1190	0.0799
		Avg. Precision	0.8600	0.8400	0.8598
		Testing time (s)	66.15	40.32	35.73

were used for solving the partial binary classification problems. Usually, the most important criterion when evaluating a classifier is its prediction performance, but very often the testing time of the classifier can be equally important. In our experiments, four different multi-label classification problems were addressed by

each classifying methods. The recognition performance and the testing time were recorded for every method. The problems considered in the experiments include scene [11] (scene), gene function [12] (yeast), text [13](enron) and music [14] (emotions) classification.

The complete description of the datasets (domain, number of training and test instances, number of features, number of labels) is shown in Table 1.

In all classification problems the classifiers were trained using all available training samples of the sets and were evaluated by recognizing all the test samples from the corresponding set. Table 2 gives the performance of each method applied on each of the datasets. The first column of the table describes the datasets. The second column shows the values of the threshold t for each dataset separately, for which the presented results of TSVA are obtained.

The value of the threshold t for each dataset was determined by 5-fold cross validation using only the samples of the training set in order to achieve maximum benefit in terms of prediction results on testing speed.

Table 2 clearly shows that among the three tested approaches TSVA offers best performance in terms of testing speed. The results show that for the four treated classification problems TSVA is 2 to 4 times faster than calibrated label ranking algorithm with majority voting and 10% to 15% faster than the QWeightedML method. It can also be noticed that TSVA offers better performance than QWeightedML method in all evaluation metrics, while showing comparable performance to calibrated label ranking algorithm with majority voting. The dependence of the predictive performances for different values of the threshold t ($0 \leq t \leq 1$) are shown on Fig. 2. Fig. 3 shows the testing time

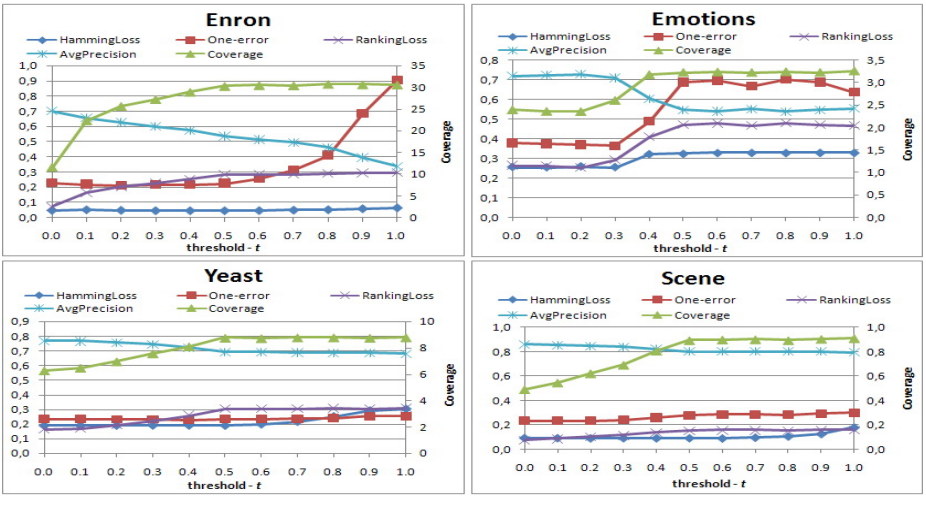


Fig. 2. Predictive performance of TSVA as a function of the threshold t ($0 \leq t \leq 1$) for each dataset

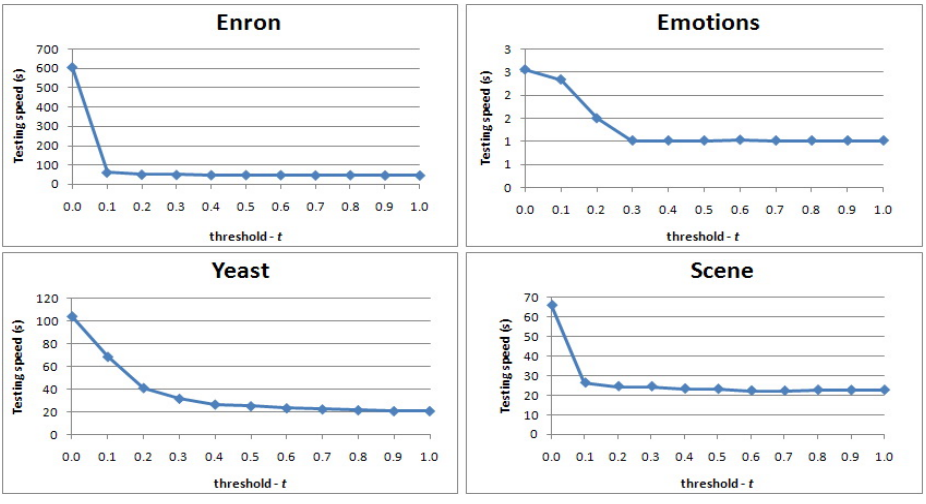


Fig. 3. Testing time of TSVA as a function of the threshold t ($0 \leq t \leq 1$) for each dataset measured in seconds

for the four classification problems as a function of the selected threshold t. It can be noticed that for small values of the threshold t (0.0 - 0.2) the predictive performance of TSVA changes moderately, but the testing time decreases for more than 40%. The reduction of the testing time of the TSVA over the CLR-S becomes even more notable as the number of labes in the treated classification problem increases. The experiments showed that for the enron dataset with quite big number of labels (53) the testing time of TSVA is four times shorter comparing to the calibrated label ranking algorithm.

5 Conclusion

A two stage voting architecture (TSVA) for efficient pair-wise multiclass voting to the multi-label setting was presented. The performance of this architecture was compared with the calibrated label ranking method with majority voting strategy for pair-wise multi-label classification and the QWeightedML algorithm on four different real-world datasets (enron, yeast, scene and emotions). The results show that the TSVA significantly outperforms the calibrated label ranking method with majority voting and the QWeightedML algorithm in term of testing speed while keeping comparable or offering better prediction performance. TSVA was 2 to 4 times faster than calibrated label ranking algorithm with majority voting and 10% to 15% faster than the QWeightedML method. TSVA is expected to show even bigger advantage when addressing classification problems with large number of labels.

References

1. Furnkranz, J.: Round robin classification. Journal of Machine Learning Research 2(5), 721–747 (2002)
2. Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multiclass classification by pairwise coupling. Journal of Machine Learning Research 5(8), 975–1005 (2004)
3. Brinker, K., Furnkranz, J., Hullermeier, E.: A unified model for multilabel classification and ranking. In: 17th European Conference on Artificial Intelligence, Riva Del Garda, Italy, pp. 489–493 (2006)
4. Park, S.H., Furnkranz, J.: Efficient pairwise classification. In: 18th European Conference on Machine Learning, Warsaw, Poland, pp. 658–665 (2007)
5. Loza Mencia, E., Park, S.H., Furnkranz, J.: Efficient voting prediction for pairwise multi-label classification. Neurocomputing 73, 1164–1176 (2010)
6. Furnkranz, J., Hullermeier, E., Loza Mencia, E., Brinker, K.: Multi-label classification via calibrated label ranking. Machine Learning 73(2), 133–153 (2008)
7. Schapire, R.E., Singer, Y.: Boostexter: a boosting-based system for text categorization. Machine Learning 39(2), 135–168 (2000)
8. http://mulan.sourceforge.net/
9. http://www.cs.waikato.ac.nz/ml/weka/
10. http://www.csie.ntu.edu.tw/~cjlin/libsvm/
11. Boutell, M.R., Luo, J., Xipeng, S., Brown, C.: Learning multi-labelscene classiffication. Pattern Recognition 37(9), 1757–1771 (2004)
12. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. Advances in Neural Information Processing Systems 14 (2001)
13. http://bailando.sims.berkeley.edu/enron_email.html
14. Trohidis, K., Tsoumakas, G., Vlahavas, I.: Multilabel classification of music into emotions. In: International Conference on Music Information Retrieval, Philadelphia, PA, USA, pp. 320–330 (2008)

Newton Trees*

Fernando Martínez-Plumed, Vicent Estruch, Cèsar Ferri,
José Hernández-Orallo, and María José Ramírez-Quintana

DSIC, Universitat Politècnica de València, Camí de Vera s/n, 46022 València, Spain
{fmartinez,vestruch,cferri,jorallo,mramirez}@dsic.upv.es

Abstract. This paper presents Newton trees, a redefinition of probability estimation trees (PET) based on a stochastic understanding of decision trees that follows the principle of attraction (relating mass and distance through the Inverse Square Law). The structure, application and the graphical representation of Newton trees provide a way to make their stochastically driven predictions compatible with user's intelligibility, so preserving one of the most desirable features of decision trees, comprehensibility. Unlike almost all existing decision tree learning methods, which use different kinds of partitions depending on the attribute datatype, the construction of prototypes and the derivation of probabilities from distances are identical for every datatype (nominal and numerical, but also structured). We present a way of graphically representing the original stochastic probability estimation trees using a user-friendly gravitation simile.We include experiments showing that Newton trees outperform other PETs in probability estimation and accuracy.

Keywords: Probability Estimation Trees, Decision Trees, Distance Methods, Inverse Square Law, Stochastic Decision Trees.

1 Introduction

Decision tree learning [19] is one of the most popular (and powerful) techniques in machine learning and, very especially, in data mining. Two of the most important features of decision trees are their divide-and-conquer covering of the problem space and the use of decisions defined over univariate conditions (although multivariate variants exist). Decision tree learning has evolved through the introduction of datatype-specific condition schemes, dozens of splitting criteria, and many class assignment, pruning and stopping rules.

Probability Estimation Trees (PETs) [17] [6], whose output is a probability rather than a crisp decision, are heirs of this technology, and are generally preferable over classical decision trees, whenever the goal is good rankings or good probability estimation. Initially, PETs were improved by using smoothing in the leaves [17] or through a pruning-smoothing [6]. The decision tree was

* This work has been partially supported by the EU (FEDER) and the Spanish MEC/MICINN, under grant TIN 2007-68093-C02 and the Spanish project "Agreement Technologies" (Consolider Ingenio CSD2007-00022).

J. Li (Ed.): AI 2010, LNAI 6464, pp. 174–183, 2010.

unaltered and the rules which were derived from it were consistent with its predictions. However, many other recent extensions of PETs use the decision tree as a skeleton upon which a complex decision making process takes place. The way the decision tree looks and the way it must be used to obtain the predictions are no longer easy to understand or even consistent.

In an effort of getting the most from decision tree learning for probability estimation, in this paper we present a new Stochastic Probability Estimation Tree learning technique. Splits are constructed by using attribute prototypes which work as attractors, following an inverse square law using the distance to the prototype and its mass, similar to other 'gravitational' approaches in machine learning [9, 16]. We will present the details of Newton trees and we will show that they introduce a series of new features and important contributions, namely:

- We use the notion of distance in a univariate way as a general way of treating any kind of datatype (numerical, nominal, ordinal or structured).
- We construct the tree based on the principle of attraction and we derive the probabilities, use and represent the tree using the same principle.
- We handle numerical, nominal and ordinal attributes in the same way. We do not have to *type* attributes but just provide a distance for each datatype.
- We use medoids (prototypes from the set of attribute values) and not centroids, so properly handling both continuous and discrete datatypes. For continuous datatypes we only construct a cluster per attribute and class, and not a cutpoint between each pair of values. So, we reduce the number of partitions to evaluate (see Section 3.2).
- We provide a graphical representation of the trees to easily interpret them.
- We evaluate the trees using a qualitative measure of error (accuracy), a measure of ranking quality (AUC, *Area Under the ROC Curve*) and a measure of calibration and refinement quality (MSE, *Mean Squared Error*).

The paper is organised as follows. Section 2 introduces notation and basic terminology on decision tree learning and probability estimation trees, and also reviews some related work. Section 3 introduces Newton Trees, by first describing the attraction function and then explaining how trees are learned and used to obtain the probability estimations. It also introduces a user-friendly representation of Newton trees. Section 4 includes a set of experiments, which compare Newton Trees with a common PET (C4.5 without pruning and Laplace estimation). Finally, Section 5 presents the conclusions and the future work.

2 Notation and Previous Work

2.1 Notation

The set of all possible unlabelled examples E is composed of all the elements $e = \langle e_1, e_2, ..., e_m \rangle$ with m being the number of attributes. The attribute names are denoted by $\langle x_1, x_2, ..., x_m \rangle$. A labelled dataset D is a set of pairs $\langle e, i \rangle$ where

$e \in E$ and $i \in C$, where C is the set of classes. The number of classes, $|C|$, is denoted by c. We define a probability estimator as a set of c functions $p_{i \in C} : E \to \mathcal{R}$ such that $\forall i \in C, e \in E : 0 \leq p_i(e) \leq 1$ and $\forall e \in E : \sum p_{i \in C}(e) = 1$. Decision trees are formed of nodes, splits and conditions. A *condition* is any Boolean function $g : E \to \{true, false\}$. A *split* or *partition* is a set of s conditions $g_k : 1 \leq k \leq s$. A *decision tree* can be defined recursively as follows: (i) a node with no associated split is a decision tree, called a leaf; (ii) a node with an associated split $g_k : 1 \leq k \leq s$ and a set of s children t_k, such that each condition is associated with one and only one child, and each child t_k is a decision tree, is also a decision tree. Given a node ν, $Children(\nu)$ denotes the set of its children and $Parent(\nu)$ denotes its predecessor node. The special node where $Parent(\nu) = \emptyset$ is called the *root* of the tree. After the training stage, the examples will have been distributed among all the nodes in the tree, where the root node contains all the examples and downward nodes contain the subset of examples that are consistent with all its ancestors' conditions. Therefore, every node has particular absolute frequencies $n_1, n_2, ..., n_c$ for each class. The cardinality of the node is given by $\sum n_i$. A *decision tree classifier* is defined as a decision tree with an associated labelling of the leaves with classes. A PET is a decision tree which outputs a probability for each class.

2.2 Related Work

Existing Probability Estimation Trees output a probability but are not necessarily probabilistic in nature. A first issue is that they typically use a divide-and-conquer philosophy for constructing the tree but the same philosophy is used to make a prediction. Given an example, a sequence of decisions will lead to a leaf of the tree where a value is returned (a class in classification trees, a number in regression trees, a probability in PETs, etc.). The rest of the information of the tree is wasted (although there are exceptions [4,6,14]). In decision theory, though, this crisp view of decisions is awkward, since each decision can have an associated probability, and the overall probability must be computed by considering the whole structure of the tree. This kind of tree are frequently (but not always) called stochastic decision trees (e.g. [12]).

A second issue is that this use of all the paths in the tree can be made in such a way that the probabilities of the tree are independent to the instance which is being processed. In fact, this has been the approach in [14], by using an ad-hoc parameter which is used to determine the probability of each child in a partition. More recent approaches ([1], [2]) have made the probability depend on the proximity to the cut-point for the attribute, by using Kernel Density Estimates. In other words, a tree can be constructed by a classical algorithm (such as C4.5 [18] or CART [3]) and its probabilistic or stochastic interpretation can be inconsistent to the way the decision tree was constructed.

A third issue is how different datatypes are handled. Many of the previous approaches only deal with numerical attributes ([1], [2]) or only with nominal attributes. When handling both, the trees just preserve the very specific way of handling numerical attributes with cutpoints and nominal attributes with

equalities, as C4.5 [18] or CART [3]. Even in the case of fuzzy decision trees (which often provide a more integrated view of nominal and numerical attributes) it is unclear how decision trees can be applied to problems where some attributes are from other (structured) datatypes such as intervals, sequences or sets.

Having all the previous approaches to PETs, in this work we propose a new decision tree learning method which has been designed from scratch with the goals of being stochastic in nature, general and flexible in the way it handles data attributes, and intelligible.

3 Stochastic Distance-Based Probability Estimation Trees

In this section we define our Stochastic Probability Estimation Tree learning technique which leads to Newton trees.

3.1 Gravitational Partitions

When constructing splits, decision trees typically generate conditions which are then evaluated to see how well they separate the classes. Instead of that, we propose to define a node/cluster per class and then try to find the characterisation of each node in terms of one attribute at a time (univariate).

Following this idea, one first approach is to use Kernel Density Estimation [21] in order to derive a probability density function (*pdf*), from the examples belonging to each class. However, many of these techniques will construct a parametrised or composite *pdf* that will make partitions unintelligible, apart from having the risk of overfitting. Another approach is to derive a prototype for each node, and then, to derive a probability from the prototypes. In order to treat discrete datatypes appropriately, we use a medoid (the element in each cluster such that its average distance to the rest is the lowest). If we generate prototypes, one possibility to derive probabilities from them is to assume some probability distribution. For instance, if we consider a normal distribution for each node with centre at the prototype and with standard deviation equal to the mean of distances of the elements of the node, we have a *pdf*. Figure 1 (left) shows the *pdf* using a Gaussian with centres 3 and 8, with standard deviations 1 and 3.5 (respectively) and masses 20 and 100 (respectively). This can be converted into conditional probabilities by mere normalisation, as shown in Figure 1 (right).

The problem of the previous approach is that when masses are too disparate, one distribution can cover the other, giving a plain partition where all the elements go to one prototype. One criterion to avoid this is to give extra importance to distance, so that at distance 0 the probability is always 1. A way to do this is to employ an inverse-square law such as in gravitation. Hence, we define the following *attraction* function between an element e of mass m_e (we will assume $m_e = 1$) and a prototype π of mass m_π separated by a distance $d(e, \pi) = d$:

$$attraction(e, \pi) = \frac{m_e m_\pi}{d(e, \pi)^2} = \frac{m_\pi}{d^2}$$

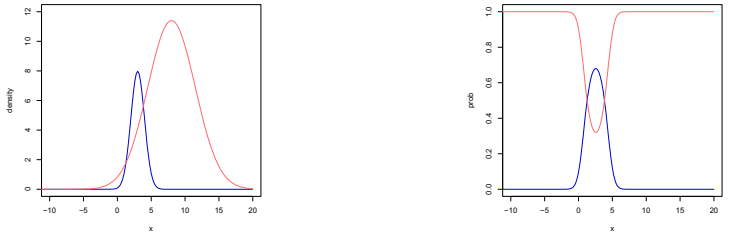


Fig. 1. (Left) Two normal distributions placed at centres 3 and 8, with standard deviations 1 and 3.5 (respectively) and masses 20 and 100 (respectively). (Right) The conditional probabilities derived from the Gaussians.

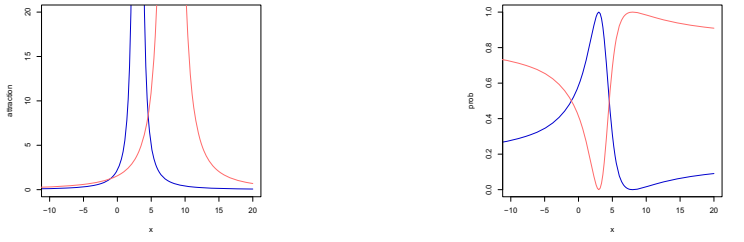


Fig. 2. (Left) Two gravitational centres at 3 and 8 with masses 20 and 100 (respectively). (Right) The probabilities derived from the gravitational centres.

We are interested in deriving class probabilities by considering this attraction. Figure 2 shows the attraction (left) and the probability (right) with the same parameters as before (note that the standard deviation is no longer used).

An interesting property is that when the distance goes to infinity the probabilities tend to converge to the mass proportion. For instance, if we have two centres at 3 and 8, and 8 has much more mass (as in the previous example), it is easy to see that the attraction to 8 will be higher than the attraction to 3 for a point placed at -100.

Of course, the idea of using the gravitational law in machine learning is not new at all, for instance in clustering [9] or classification [16]. The same Inverse Square Law principle is presented in some variants of Kernel Density Estimation, several classification techniques such as weighted kNN, where the weight is a kernel which is simply defined as the inverse of the distance, or in some other clustering algorithms. To our knowledge, its use for decision trees is new.

3.2 Tree Generation

Centre splitting [20] is a machine learning method which consists in dividing the input space in different regions where each region is represented by a centre[1]. In every iteration, a centre is calculated for every different class which is presented in the area. Then, every example is associated to its nearest centre. This process

[1] The centre may match to an existing example or not.

is repeated until the area is pure. One of the special features of this method is that examples are managed as a whole.This appreciation leads us to propose a decision tree inference strategy where centroids are computed by considering only the values of one attribute, which allows us to join centre splitting and decision tree learning techniques in an elegant way.

The detailed definition of the algorithm can be found in [15]. Here, we give a more sketchy description: for each attribute x_r and for each class i, a prototype $\pi_{r,i}$ is calculated as the attribute value with lowest mean distance to the elements of the class. Once this process is finished, the splitting attribute is selected according to one of the well-known splitting criteria (for instance, gain ratio [18]). Then, the split proceeds by associating every instance to its closest attribute prototype, which typically produces impure clusters[2]. Although the computation of distances is quadratic on the number of instances, we can reduce it by using a distance matrix per attribute (of size $n_r \times n_r$, where n_r is the number of different attribute values) prior to the algorithm execution. But, more importantly, if we have m attributes and n_r values per attribute, we only construct (and evaluate) $O(m)$ partitions and not $O(n_r \times m)$, the typical order for classical decision tree learning algorithms using midpoints for continuous attributes.

3.3 Stochastic Probability Calculation

Now, we illustrate how a Newton Tree is used to estimate probabilities in a stochastic way. In what follows, $\overrightarrow{p}(\nu, e) = \langle p_1(\nu, e), \ldots, p_c(\nu, e)\rangle$ denotes the probability vector of example e at node ν, where $p_i(\nu, e)$ denotes the probability that e belongs to class i at node ν. With $\hat{p}(\nu, e)$ we denote the probability that e falls into node ν (coming from its parent), which is derived from the attraction that ν exert over e, that is $\hat{p}(\nu, e) = \frac{attraction(e,\nu)}{\sum_{\mu \in Children(Parent(\nu))} attraction(e,\mu)}$.

Given a new example e and a Newton tree T, the objective is to calculate the probability vector at the root of T, $\overrightarrow{p}(root, e)$. Basically, the idea is to compute downwards the probability of falling in each leaf, calculate the leaf probability vector and then to propagate upwards the leaf probability vector to the root to obtain the total class probability vector $\overrightarrow{p}(root, e)$. The leaf probability vectors can be obtained once the tree T has been built by applying Laplace correction as has been shown in [17,6]. For each example, we calculate the probability of choosing each child node μ if placed at the parent node ν using the attraction (i.e., $\hat{p}(\mu, e)$). This probability is multiplied by the probability vector of the child ($\overrightarrow{p}(\mu, e)$):

Definition 1. Stochastic Probability Vector Estimation
Given an example e and a Newton tree T, the probability vector $\overrightarrow{p}(root, e)$ at the root of T is estimated by applying

$$\forall \nu \in T : \overrightarrow{p}(\nu, e) = \begin{cases} \sum_{\mu \in Children(\nu)} \hat{p}(\mu, e) \cdot \overrightarrow{p}(\mu, e) & \text{if } \nu \text{ is not a leaf} \\ \langle Laplace(1, \nu), \ldots, Laplace(c, \nu)\rangle & \text{if } \nu \text{ is a leaf} \end{cases}$$

[2] Note that, during the splitting process, we apply the *attraction* function assuming that the mass is the unit. This is due to the fact that the total mass of a node is not known until all the instances have been associated to its prototype.

where $Laplace(j, \nu)$ *is the Laplace correction of the frequency of elements of class* j *in node* ν.

The stochastic calculation of the probabilities seen above may seem too cryptic for a general use of these trees if intelligibility is a requirement. In order to address this issue, we show a graphical representation of Newton trees, which may help users understand how the stochastic probability assignment is made, and to get insight from the tree.

Figure 3 (left) shows this user-friendly representation of a Newton Tree for the Hepatitis dataset from the UCI repository [8]. Note that all partitions are binary because this is a two-class problem, namely DIE and $LIVE$. The two first splits are made over the numerical attributes $PROTIME$ and $ALK\_PHOSPHATE$, respectively, and the other two splits are made over the nominal attributes SEX and $FATIGUE$. The nodes are represented as balls of a size which is proportional to the node mass (for instance, the node with a mass of 17 represents that 17 training examples fall into it). The ball also shows the proportion of examples of each class in different colours. Additionally, the value for the attribute prototype is shown in the middle of each ball. Finally, the smoothed probabilities per class at the leaves are also provided (in the figure, as a small table below each leaf). In order to ease the understanding on how probabilities are derived, Figure 3 (right) shows

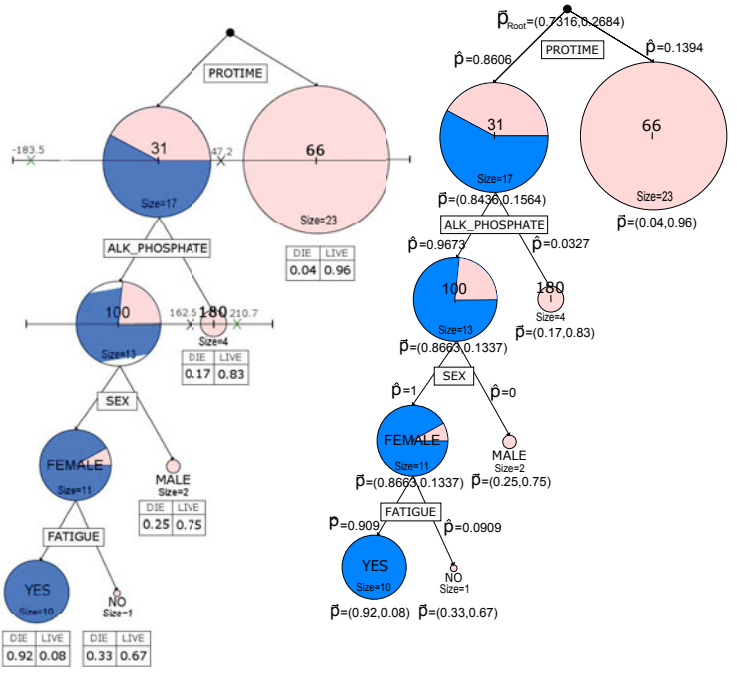


Fig. 3. (Left) Newton Tree for the hepatitis dataset. (Right) The node probability vectors, children probabilities and global probability vector for example (PROTIME=40, ALK_PH=120, SEX=FEMALE, FATIGUE= UNKNOWN).

the internal probabilities (vectors and node probabilities) and the top vector probability for example ($PROTIME = 40; ALK\_PHOSPHATE = 120; SEX = FEMALE; FATIGUE = UNKNOWN$), which is $(0.7316, 0.2684)$, a relatively clear DIE case. All these graphical elements in the Newton Trees representation may help users understand the way in that probabilities are estimated, making Newton trees less cryptic than other PET methods.

4 Experiments

The aim of this section is to compare Newton trees with a common implementation of Probability Estimation Trees, namely unpruned decision trees with Laplace smoothing in the leaves as suggested by [17] [6]. In particular, we chose J48 (the variant of C45.) implemented in Weka [10]. We used *Gain ratio* as splitting criterion for Newton trees and J48. The evaluation was performed over 30 datasets from the UCI repository [8], from which we removed instances with missing values (see [15] for their characteristics). We set up a 20×5-fold cross validation, making a total of 100 learning runs for each pair of dataset and method (3,000 overall). As evaluation metrics we used ([7]): accuracy, as a qualitative measure of error, AUC (*Area Under the Curve*) as a measure of ranking

Table 1. Comparison between Newton trees and unpruned J48 with Laplace correction

Name	Classes	Att Type	Newton Trees			Unpruned Laplace J48		
			Acc.	AUC	MSE	Acc.	AUC	MSE
anneal	6	Mixed	97.5110	**0.8943**	0.0119	**98.7800**	0.8890	**0.0073**
autos_5c	5	Mixed.	**79.5060**	**0.9043**	0.0825	77.7130	0.8827	0.0840
balance-scale	3	Num.	**79.5520**	0.7962	0.1050	78.6880	**0.8199**	**0.0998**
breast-cancer	2	Nom.	**73.0110**	**0.6436**	**0.1929**	67.9360	0.6084	0.2233
chess-kr-vs-kp	2	Nom.	98.5050	0.9975	0.0135	**99.3050**	**0.9988**	**0.0064**
cmc	3	Mixed.	**50.1720**	**0.6739**	**0.2025**	49.1100	0.6658	0.2107
credit-a	2	Mixed.	**84.9310**	**0.9107**	**0.1118**	82.7960	0.8982	0.1256
credit-g	2	Mixed.	**70.3300**	**0.7202**	**0.1897**	68.2900	0.7016	0.2159
diabetes	2	Num.	71.8630	0.7801	**0.1798**	**72.8070**	0.7772	0.1877
glass	7	Num.	67.2940	0.7828	0.0901	67.0340	0.7895	0.0879
heart-statlog	2	Num.	**78.0740**	**0.8626**	**0.1490**	76.1850	0.8398	0.1753
hepatitis	2	Mixed.	**83.4370**	**0.7570**	**0.1143**	79.4370	0.6542	0.1498
ionosphere	2	Num.	88.9160	0.9235	0.0916	88.8460	0.9195	0.0917
iris	3	Num.	**94.7660**	**0.9938**	**0.0315**	94.0330	0.9710	0.0349
monks1W	2	Nom.	93.5230	**0.9899**	0.0606	92.7690	0.9761	**0.0519**
monks2W	2	Nom.	**85.8750**	**0.9378**	**0.1124**	61.3790	0.6456	0.2348
monks3W	2	Nom.	98.6730	**0.9926**	0.0166	98.6370	0.9909	**0.0135**
mushroom	2	Nom.	99.9910	0.9999	0.0193	100.0000	1.0000	**0.0001**
new-thyroid	3	Num.	92.6970	**0.9854**	**0.0438**	92.3480	0.9237	0.0454
pimaW	2	Num.	71.8630	0.7801	**0.1798**	**72.7750**	0.7772	0.1877
sonar	2	Num.	**77.5990**	**0.8499**	**0.1538**	73.3710	0.7888	0.2162
soybean	19	Nom.	89.2420	0.9771	0.0228	**91.2270**	0.9770	**0.0183**
spectf_train	2	Num.	67.3120	0.7301	0.2097	**71.7500**	0.7365	0.2196
tae	3	Mixed.	**58.7010**	**0.7398**	**0.1877**	54.1660	0.7078	0.1996
tic-tacW	3	Nom.	78.1110	0.8526	0.1426	**79.3990**	**0.8699**	**0.1393**
vehicle3c	3	Num.	72.1210	0.8441	0.1355	**73.0240**	**0.8807**	**0.1251**
vote	2	Nom.	94.5020	**0.9892**	0.0383	**95.1370**	0.9827	**0.0355**
vowel	11	Mixed.	75.3580	**0.9671**	0.0578	**79.5400**	0.9157	**0.0447**
wine	3	Num.	**94.3840**	**0.9905**	**0.0408**	92.2070	0.9544	0.0471
zoo	7	Mixed.	**94.9020**	**0.7243**	0.0252	93.1610	0.7147	**0.0234**
Mean (All)			82,0907	0,8664	0,1004	80,7283	0,8419	0,1101
Mean ($c = 2$)			83,6503	0,8665	0,1146	81,3388	0,8310	0,1334
Mean ($c > 2$)			80,3084	0,8662	0,0843	80,0307	0,8544	0,0834
Mean (Nominal)			90,1592	0,9311	0,0688	87,3099	0,8944	0,0803
Mean (Numerical)			79,7034	0,8599	0,1175	79,4223	0,8482	0,1265
Mean (Mixed)			77,2053	0,8102	0,1093	75,8881	0,7811	0,1179

Table 2. Aggregated results using the statistical tests

Unpruned Laplace J48 / Netwon Trees	Acc.	AUC	MSE
All	14/6/10	18/8/4	14/4/12
Nominal	2/3/4	5/2/2	2/0/7
Numerical	5/3/4	5/5/2	7/3/2
Mixed	7/0/2	9/0/0	5/1/3

quality, (using Hand & Till's multiclass version [11]) and MSE (*Mean Squared Error*) as a measure of calibration and refinement quality.

Table 1 shows the average accuracy, AUC and MSE obtained by the two algorithms. At the bottom, we also show the mean values for all the datasets. These means are just illustrative. To analyse whether the differences are significant, we used the Wilcoxon signed-ranks test with a confidence level of $\alpha = 0.05$ and $N = 30$ data sets, as suggested in [5]. Significant differences are shown in bold. Finally, in Table 2 we focus on these differences, showing an entry $w/t/l$ for each measure and dataset subset, which indicates that Newton trees win in w, tie in t, and lose in l datasets, compared to the J48 PETs. From the tables, we see that Newton trees outperform J48 PETs in the three measures (Accuracy, AUC and MSE), and with the means in Table 1, in any selection depending on the type of dataset (multiclass/binary, nominal/numerical/mixed). The strongest differences are found in AUC, which is the recommended measure when evaluating PETs ([13]). If we look at the significance results in Table 2, we have a similar picture. The exception is the result for nominal datasets. While AUC is still much better, the results in MSE are worse (and as a result so is accuracy). This indicates a bad calibration of the results for datasets with only nominal partitions, which might be caused by the way discrete distances affect on the attraction measure, although more research should be done to clarify this (since there are only 7 datasets in this subset).

5 Conclusions and Future Work

This paper has presented a novel probability estimation tree learning method which is based on computing prototypes and applying an Inverse Square Law that uses the distance to the prototype and its mass, in order to derive an attraction force which is then converted into a probability. The trees can be graphically represented in such a way that their meaning and patterns can be understood. The use of prototypes (mediods) instead of centroids allows for the use of our trees for any kind of datatype (continuous or discrete), as long as we provide a distance function for each datatype. Consequently, we can apply our trees to structured datatypes, such as sequences, sets, ordinal data, intervals or even images and texts. More importantly, we can use the tree with a mixture of all these datatypes. If distance matrices are preprocessed (only once for each attribute before start), the computation of the prototypes is much more efficient than the split population schemes in traditional decision trees, since we group by classes and then compute the mediod of each cluster. Consequently, the number of different splits to evaluate at each node is equal to the number of attributes and does not depend on midpoints or the size of the dataset.

There are many research lines to pursue. One is to use the mass also when constructing the tree or using all the attribute values as possible clusters. However, these two modifications would entail extra computational cost and could only be justified if there is a significant improvement in the results.

References

1. Alvarez, I., Bernard, S.: Ranking cases with decision trees: a geometric method that preserves intelligibility. In: IJCAI, pp. 635–640 (2005)
2. Alvarez, I., Bernard, S., Deffuant, G.: Keep the decision tree and estimate the class probabilities using its decision boundary. In: IJCAI, pp. 654–659 (2007)
3. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Chapman & Hall, New York (1984)
4. Buntine, W.: Learning classification trees. Stats. and Computing 2(2), 63–73 (1992)
5. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
6. Ferri, C., Flach, P., Hernandez-Orallo, J.: Improving the auc of probabilistic estimation trees. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) ECML 2003. LNCS (LNAI), vol. 2837, pp. 121–132. Springer, Heidelberg (2003)
7. Ferri, C., Hernández-Orallo, J., Modroiu, R.: An experimental comparison of performance measures for classification. Pattern Recogn. Lett. 30(1), 27–38 (2009)
8. Frank, A., Asuncion, A.: UCI Machine Learning Repository (2010)
9. Gomez, J., Dasgupta, D., Nasraoui, O.: A new gravitational clustering algorithm. In: Int. Conf. on Data Mining. Society for Industrial & Applied, p. 83 (2003)
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software. SIGKDD Explorations 11(1), 10–18 (2009)
11. Hand, D.J., Till, R.J.: A simple generalisation of the area under the ROC curve for multiple class classification problems. Machine Learning 45(2), 171–186 (2001)
12. Hespos, R.F., Strassmann, P.A.: Stochastic decision trees for the analysis of investment decisions. Management Science 11(10), 244–259 (1965)
13. Huang, J., Ling, C.X.: Using auc and accuracy in evaluating learning algorithms - appendices. IEEE Trans. Knowl. Data Eng. 17(3) (2005)
14. Ling, C.X., Yan, R.J.: Decision tree with better ranking. In: International Conference on Machine Learning, vol. 20(2), p. 480 (2003)
15. Martinez-Plumed, F., Estruch, V., Ferri, C., Hernández-Orallo, J., Ramírez-Quintana, M.J.: Newton trees. extended report. Technical report, DSIC, UPV (2010), http://www.dsic.upv.es/~flip/NewtonTR.pdf
16. Peng, L., Yang, B., Chen, Y., Abraham, A.: Data gravitation based classification. Information Sciences 179(6), 809–819 (2009)
17. Foster, J.: Provost and Pedro Domingos. Tree induction for probability-based ranking. Machine Learning 52(3), 199–215 (2003)
18. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993)
19. Rokach, L., Maimon, O.: Data Mining with Decision Trees: Theory and Applications. World Scientific, Singapore (2008)
20. Thornton, C.J.: Truth from trash: how learning makes sense. The MIT Press, Cambridge (2000)
21. Turlach, B.A.: Bandwidth selection in kernel density estimation: A review. In: CORE and Institut de Statistique (1993)

Learning Parse-Free Event-Based Features for Textual Entailment Recognition

Bahadorreza Ofoghi[1,2] and John Yearwood[1]

[1] Centre for Informatics and Applied Optimization, University of Ballarat
P.O. Box 663, Ballarat Victoria 3350, Australia
[2] Institute of Sport, Exercise, and Active Living, Victoria University
P.O. Box 14428, Melbourne Victoria 8001, Australia

Abstract. We propose new parse-free event-based features to be used in conjunction with lexical, syntactic, and semantic features of texts and hypotheses for Machine Learning-based Recognizing Textual Entailment. Our new similarity features are extracted without using shallow semantic parsers, but still lexical and compositional semantics are not left out. Our experimental results demonstrate that these features can improve the effectiveness of the identification of entailment and no-entailment relationships.

1 Introduction

Recognizing Textual Entailment (RTE) has now become a direction of study for the members of the natural language processing community and is formally described as automatically recognizing the relationship between a *hypothesis* and a *text*. The hypothesis (H) is a succinct piece of text and the text (T) includes a few sentences the meaning of which may or may not entail the truth/falsity of H. If the truth of H can be inferred from the evidence in T, then the relationship is denoted by $T \rightarrow H$. For instance, given the following:

text. *Children as young as six are being sexually abused by UN peacekeepers and aid workers, says a leading UK charity.*

hypothesis. *UN peacekeepers abuse children.*

the relation $T \rightarrow H$ holds. In this paper, we consider 2-way RTE where the two classes are: i) *Entailment*: where $T \rightarrow H$ and ii)*No-Entailment*: where either $T \rightarrow \neg H$ or there is not enough evidence available in the text to decide whether $T \rightarrow H$ or $T \rightarrow \neg H$.

A number of approaches to RTE have been developed during recent years. Systems that use morphological and lexical variations [25,23,1], classical or plausible logic [13,20,24], syntactic dependency trees [12,22,21], paraphrase detection [26], and semantic roles [25] can be named. Some RTE systems take Machine Learning (ML) procedures in order to learn classifiers that can distinguish between different entailment classes [10,5,7,14,15,2].

Different types of lexical and semantic information have been recently used for RTE. The event or status-based Frame Semantics [8,17] encapsulated in

J. Li (Ed.): AI 2010, LNAI 6464, pp. 184–193, 2010.

FrameNet [9] has been used in a few previous attempts at RTE [3,4,27,28]. FrameNet is a lexical semantic resource being developed at UC Berkeley. It encapsulates the concept of continuities between language and human experience into semantic *frames*. A semantic frame represents an event or state that contains a number of participant roles known as *frame elements* (FEs). The structure of FrameNet allows for the meaning of a single word to be dependent on the essential knowledge related to that word.

The major drawback of current RTE systems that use FrameNet information is their reliance on shallow semantic parsers that add semantic information to texts. The state-of-the-art shallow semantic parsers that can add frame semantic information to texts do not have high levels of accuracy [19]. When incorrect/insufficient semantic information is added to texts by shallow semantic parsers, systems relying on such parsers are disadvantaged significantly (Ofoghi et al. [6] demonstrate this effect on natural language Question Answering systems). To overcome this problem, we propose a semantic parsing-free approach based on ML techniques the novelty of which is two-fold:

- Unlike the works in [3,4,27], our method does not rely on other challenging learning procedures for word sense disambiguation and semantic role labeling.
- Unlike the work in [28], we utilize more sophisticated FrameNet-based features along with a number of other well-known lexical semantic features and ML techniques to measure the extent to which event-based similarity features can improve RTE effectiveness.

2 Feature Space Engineering

To learn an automated classification system for identifying entailment relationships, we extract a number of lexical, syntactic, semantic, and parse-free event-based features from both texts and hypotheses. Prior to feature extraction, all text and hypothesis terms are lemmatized using the *TreeTagger* lemmatizer [16].

2.1 Lexical and Syntactic Features

The lexical features (*lex*) that we use are the total number of exact terms that match between the text and hypothesis and also the least common subsequence (LCS) of text and hypothesis terms. These features are among *similarity-based* features explained in [4].

The syntactic feature (*syn*) in our work is the LCS between dependency trees of text and hypothesis sentences extracted using *Link Grammar Parser (LGP)* [11]. This feature captures the similarity of texts and hypotheses in terms of their syntactic structures and is still sensitive to structural paraphrasing.

2.2 Semantic Features

The semantic features (*sem*) that we utilize are extracted by using WordNet lexical ontology. These features include:

– **Synonyms**: The total number of synonym terms that match between the text and hypothesis term sets. This feature can overcome lexical paraphrasing.
– **Hypernyms/Hyponyms**: The total number of hypernyms and hyponyms that match between the text and hypothesis term sets. The analysis of hypernym and hyponym matching takes a *directional* approach. The procedures is formulated in Equation 1 where $s_{h/t}$ is the set of hypernyms or hyponyms for the hypothesis/text and $n_{h/t}$ represents the total number of terms t_i in the hypothesis/text. Hypernyms and hyponyms are extracted up to three links in WordNet. The idea behind using this directional approach is that WordNet hyponyms entail WordNet hypernyms e.g. "*female person*" lexically entails "*person*" but not vice versa.

This feature is designated to overcome the problem related to texts and hypotheses formulating concepts at different levels of conceptual abstraction. For instance, using this feature, it is possible to recognize the entailment relationship between "*Jack was in a European country last year.*" and "*Jack was in France last year.*" since "France" is a "European country" according to WordNet.

$$\begin{aligned} &hyper/hypo\_score(h,t) = |s_h \cap s_t| \\ &s_h = \bigcup_{i=1}^{n_h} hypernyms(t_i),\ s_t = \bigcup_{i=1}^{n_t} hyponyms(t_i) \end{aligned} \tag{1}$$

– **Antonyms**: The antonym score is calculated using Equation 2 where $s_{h/t}$ is the set of exact terms or antonyms for the hypothesis/text term t_i. A similar attribute has been used as a trigger-based feature in [4]. This feature can capture indications of contradiction or no-entailment relationships.

$$ant\_score(h,t) = |s_h \cap s_t|,\ s_h = \bigcup_{i=1}^{n_h} t_i,\ s_t = \bigcup_{i=1}^{n_t} antonyms(t_i) \tag{2}$$

– **Antonyms/Hyponyms**: This feature is measured using Equation 3 where $s_{h/t}$ is the set of hyponyms or antonyms for the hypothesis/text term t_i. This feature extends the last feature (antonyms) by looking for the occurrences of the antonyms of more specific terms of hypotheses in texts.

$$ant/hypo\_score(h,t) = |s_h \cap s_t|,\ s_h = \bigcup_{i=1}^{n_h} hyponyms(t_i),\ s_t = \bigcup_{i=1}^{n_t} antonyms(t_i) \tag{3}$$

2.3 Event-Based Features

We make use of FrameNet to extract two types of event-based features, namely *ebf* and *inter_ebf*. The ebf feature is the total number of FrameNet frames that are evoked both by text and hypothesis terms. To measure this, we use Equation 4 where the set of all frames that contain each hypothesis/text term is created by a term look-up procedure in FrameNet XML database. The union set of all framesets for all hypoesis/text terms is then created. The cardinality of the intersection of the two union sets is the score assigned to ebf. This method

does not rely on any shallow semantic parser and/or word sense disambiguation procedure to evoke FrameNet frames; instead, is only based on fetching the names of frames that contain certain terms.

Using ebf, our RTE system can relate a hypothesis and a text that share semantics at the level of an event or state (see Table 1). This type of *scenario-based* similarity may not be captured using other types of lexical resources.

$$ebf\_score(h,t) = |s_h \cap s_t|,\ s_{h/t} = \bigcup_{i=1}^{n_{h/t}} frameset(t_i),\ frameset(t_i) = \bigcup_j frame_j \quad (4)$$
$$\{\exists t \in frame_j(termset) \mid t_i = t \wedge frame_j \in FN\_frames\}$$

Table 1. An example of calculating the ebf similarity feature between the hypothesis "*Jack, a famous teacher, is Stacy's son*" and the text "*Stacy is Jack's mother*"

Term	Frames	Union of frames	Intersection of framesets
ht=jack	–		
ht=stacy	–	$hFrames = \{fame,$	
ht=famous	fame	$education\_teaching,$	
ht=teacher	education_teaching	$kinship\}$	
ht=son	kinship		$\lvert hFrames \cap tFrames\rvert =$
tt=stacy	–		$\lvert\{kinship\}\rvert$=1
tt=jack	–	$tFrames = \{kinship\}$	
tt=mother	kinship		

FrameNet frames are inter-related using a number of frame-to-frame relations explained in detail in [18]. We use the *inheritance*, *subframe*, *using*, *inchoative-of*, *causative-of*, *precedes*, and *perspective-on* relations to extract one feature per relation type that represents another level of event/state-based textual similarity. We refer to this set of features as inter_ebf. For each type of frame-to-frame relation, we extract immediately inter-related FrameNet frames to those frames evoked for each hypothesis/text term. A similar approach to the calculation of ebf is used to measure the overlap between inter-related frames evoked. Using inter_ebf, the system is able to recognize wether a hypothesis formulates a scenario that is, for instance, part of a big picture scenario or event (by measuring the overlap between inter-related frames according to the *inheritance* relationship).

In general, although our features are evoked on a term-basis procedure, the nature of the features and linguistic resources that we have used ensure that compositional semantics are indirectly taken into consideration.

2.4 Other Features

We use the task (*tsk*) in which the pair is categorized by the Text Analysis Conference (TAC). The tsk feature takes one of the seven distinct categories included

in the TAC datasets[1] and can be extracted in both development (training) and test (un-annotated) datasets. Our results without using the tsk feature show the same trend as those discussed later in this paper with using the tsk feature.

3 Empirical Analysis

3.1 Data

We use labeled hypothesis and text pairs provided by the TAC. This includes the rte 1 to 5 test sets for testing and the rte 5 development set used for training.

3.2 Experiments

We carry out two experiments for analyzing i) the overall effectiveness of our RTE system and ii) the contribution of event-based features, extracted without using shallow semantic parsers, to effective textual entailment recognition.

The results presented in the following sections have been tested for their statistical significance and the superscript symbols $*$, †, and ‡ show the statistical significance with $p < 0.25$, $p < 0.05$, and $p < 0.0001$ respectively. The statistical paired t-test has been used unless otherwise specified.

Overall system analysis. We ran a number of learning classification systems on the whole set of features explained in section 2 as well as on different subsets of the whole feature set. For simplicity, in this section we only report our system's results with the whole feature set (*/all*) and with the best subset of the features (*/bst*) that have resulted in the highest performance using the Support Vector Machines (SVMs) classifier. The bst features have been arrived at using a manual trial and error wrapper procedure[2] and includes ebf, lex, sem (excluding hypernyms/hyponyms), and inter_ebf (excluding the *requires* relation), and tsk.

Table 2 summarizes the results of our RTE system obtained for each dataset. The overall accuracies of our system are far better than a hypothesized baseline with a 50% overall accuracy at the expense of a 0 recall on one class. Compared to other TAC-RTE 2009 systems, our RTE accuracy falls close to the median accuracy[3], where still significantly higher than that by the TAC-RTE 2009 system that used FrameNet [28].

Our RTE system, trained with the rte5 development set, has a decreasing performance trend over the older datasets. The reason for this is the shorter

[1] See the TAC datasets available at http://www.nist.gov/tac

[2] In the ML literature *filter* methods of feature selection disregard any application of the features and find the best representative feature set for a dataset, whereas the *wrapper* selection methods follow an objective procedure by optimizing the system performance in terms of solving a particular problem, e.g. classification.

[3] The overall statistics of the TAC-RTE 2009 (with the rte5 test set) for 55 runs submitted by 13 participant teams shows the high, median, and low 2-way classification accuracies of 73.500, 61.170, and 50.000 respectively.

Table 2. RTE system effectiveness on different data/feature sets – the statistical significance tests carried out between the accuracies obtained by *all* and *bst* features

Data	all				bst			
	Recall		Acc. (%)	ROC	Recall		Acc. (%)	ROC
	ent.	no ent.			ent.	no ent.		
rte5-dev.	0.553	0.713	63.333	0.633	0.570	0.737	65.333	0.653
rte1-test	0.458	0.605	$53.125^{\dagger}$	0.531	0.478	0.580	52.875	0.529
rte2-test	0.375	0.698	53.625	0.536	0.400	0.685	$54.250^{\dagger}$	0.543
rte3-test	0.351	0.795	56.750	0.573	0.366	0.785	$57.000^{\dagger}$	0.575
rte4-test	0.364	0.762	56.300	0.563	0.376	0.750	56.300	0.563
rte5-test	0.540	0.670	60.500	0.605	0.533	0.673	60.333	0.603

ROC = Receiver Operating Characteristic curve

texts included in these datasets compared to the lengthier texts in the newer datasets. With short texts, it is more difficult for our ML-based RTE system to extract enough meaningful information from the hypothesis and text pairs; therefore, the system has less chance in identifying correct entailment classes.

Event-based feature analysis. For this, we ran our RTE system in two satges: stage 1) removing the inter_ebf feature set from the best subset of features described in section 3.2, and stage 2) excluding the ebf features from stage 1. The results of these experiments are summarized in Table 3.

The important observation here is the significantly lower classification performances and recall values achieved in most cases after excluding inter_ebf and ebf features from the classification task. Given the variety of the features that we have used along with the event-based features, these lower performances suggest that our new parse-free event-based features, not similar to previously used FrameNet-based features, capture useful information for RTE.

Table 3. RTE system effectiveness with the *bst* features excluding certain event-based features – the statistical significance tests carried out in comparison with the accuracies of the *bst* features and show significantly lower accuracies

Data	exc. inter_ebf				exc. inter_ebf+ebf			
	Recall		Acc. (%)	ROC	Recall		Acc. (%)	ROC
	ent.	no ent.			ent.	no ent.		
rte5-dev.	0.583	0.697	$64.000^{\dagger}$	0.640	0.443	0.703	$57.333^{\dagger}$	0.573
rte1-test	0.495	0.553	$52.375^{\dagger}$	0.524	0.433	0.608	$52.000^{\dagger}$	0.520
rte2-test	0.445	0.650	$54.750^{\ddagger}$	0.548	0.345	0.690	51.750^{*}	0.518
rte3-test	0.412	0.759	$58.125^{\ddagger}$	0.586	0.320	0.795	$55.125^{\dagger}$	0.557
rte4-test	0.398	0.718	$55.800^{\dagger}$	0.558	0.322	0.790	$55.600^{\ddagger}$	0.556
rte5-test	0.573	0.617	$59.500^{\ddagger}$	0.595	0.460	0.720	$59.000^{\dagger}$	0.590

4 Classification Model Reverse Engineering

We have looked at all individual features in the best subset of features introduced in section 3.2 to see what roles they play in the classification task of the entailment relationships. We have first used the scatter-plots shown in Figure 1 drawn from all of the datasets under experiment. None of the individual features can actually be used solely for distinguishing between the relationship classes as in all plots, points are scattered over similar areas for both classes. This emphasizes the difficulty of drawing a hyperplane to effectively separate the instances (hypothesis and text pairs) of each class by considering individual features.

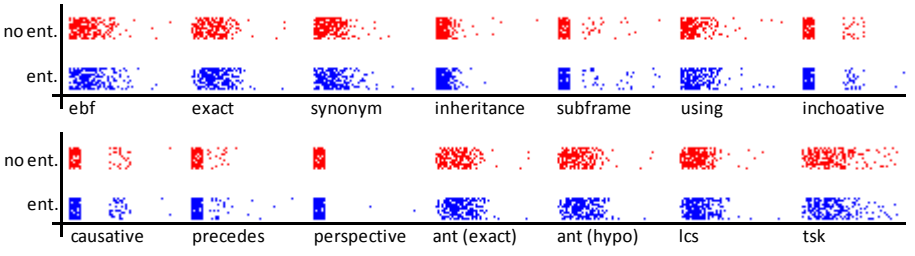


Fig. 1. Analysis of the relationships between individual features and class labels in all datasets – x-axis: features, y-axis: classes

We have then used the *Squared Correlation* (r^2) statistic ([0.0,1.0]) to find the strength of the correlation between the values of the individual features and the predicted class labels after running the classifier with the best subset of features. Figure 2 shows the results of the r^2 analysis. The *linear* r^2 statistic does not necessarily represent any *cause-effect* relationship between the features and therefore, does not suggest any ranking of the features.

The result of the r^2 analysis demonstrates that the values of *none* of the features have a strong relationship (r^2 close to 1.0) with the entailment and no-entailment classes. Because of the linearity characteristics of the r^2 analysis, this is what one might expect after looking at the scatter-plots in Figure 1.

In general, the scatter-plot and r^2 analysis of the individual features emphasizes the *non-linearity* of the relationships between the features and the classes and suggests that a sophisticated classification system is required to overcome the entailment recognition problem. This validates the ML approach we have taken as well as the non-linear learning classifier system (SVMs with a Poly Kernel) utilized in this work.

More importantly, the statistical analyses suggest that making use of combinations of the features is required for achieving high levels of RTE performances. The results shown in Table 4 suggest that even *no individual group of features* is sufficient for RTE. This again demonstrates that our parse-free event-based

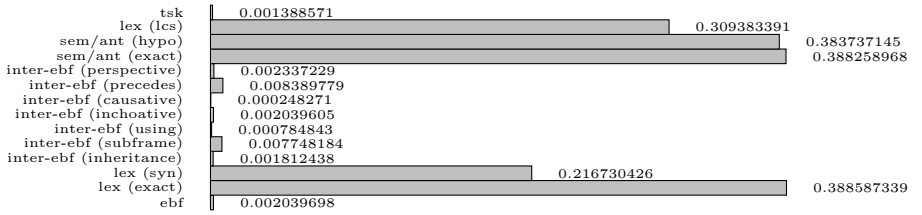


Fig. 2. The r^2 statistic between the values of the individual features and the class labels in the rte5 development dataset

Table 4. Feature group analysis on all datasets – results obtained with 10-fold cross validation – the statistical significance tests carried out using the standard t-test – the ‡s show significantly lower accuracies than the best accuracy achieved by *all* features

Feature set	Recall		Acc. (%)	ROC
	ent.	no ent.		
lex	0.384	0.707	$54.456^{\ddagger}$	0.545
sem	0.394	0.698	$54.565^{\ddagger}$	0.546
ebf+inter_ebf	0.174	0.831	$50.130^{\ddagger}$	0.503
all	0.471	0.648	55.913	0.560

features play an important role in improving the RTE effectiveness when used in conjunction with other textual features. This becomes more significant noticing that our new features remove the complexity involved in shallow semantic parsers, the current overall state-of-the-art accuracies of which are not very high.

5 Concluding Remarks

Shallow semantic parsing is a challenging task in the natural language processing domain. We have developed a RTE system to analyze the impact of event-based information encapsulated in FrameNet on the textual entailment recognition performance where no semantic parsing is carried out. Instead, we employed a shallow term lookup-based procedure for extracting event-based features. We used a ML-based approach that utilizes a variety of textual features including lexical, syntactic, and semantic features in conjunction with event-based features.

The results of our experiments demonstrate that: i) our proposed event-based features, independent from shallow semantic parsing, play an important role in significantly enhancing the effectiveness of RTE and ii) a combination of different types of features, including the proposed parse-free event-based ones, is required in ML-based RTE to reach high levels of effectiveness.

References

1. Rodrigo, A., Penas, A., Verdejo, F.: Towards an entity-based recognition of textual entailment. In: Proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment, Gaithersburg, USA (2008)
2. Ageno, A., Farwell, D., Ferres, D., Cruz, F., Rodriguez, H.: TALP at TAC 2008: A semantic approach to recognizing textual entailment. In: Proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment, Gaithersburg, USA (2008)
3. Burchardt, A., Pennacchiotti, M.: FATE: a FrameNet-Annotated Corpus for Textual Entailment. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech, Morocco (2008)
4. Burchardt, A., Pennacchiotti, M., Thater, S., Pinkal, M.: Assessing the impact of frame semantics on textual entailment. Natural Language Engineering 15(4), 527–550 (2009)
5. Hickl, A., Bensley, J., Williams, J., Roberts, K., Rink, B., Shi, Y.: Recognizing textual entailment with LCC's GROUNDHOG system. In: Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment, Venice, Italy, pp. 80–85 (2006)
6. Ofoghi, B., Yearwood, J., Ma, L.: The impact of frame semantic annotation, frame alignment techniques, and fusion methods on factoid answer processing. Journal of the American Society for Information Science and Technology (JASIST) 60(2), 247–263 (2009)
7. Mac Cartney, B., Grenager, T., de Marneffe, M.-C., Cer, D., Manning, C.D.: Learning to recognize features of valid textual entailments. In: Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, New York City (2006)
8. Fillmore, C.J.: Frame semantics and the nature of language. In: Proceedings of the Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech, pp. 20–32 (1976)
9. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: Proceedings of the 17th International Conference on Computational Linguistics (COLING), pp. 86–90. Universite de Montreal, Montreal (1998)
10. Corley, C., Mihalcea, R.: Measuring the semantic similarity of texts. In: Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, Ann Arbor, MI (2005)
11. Sleator, D., Temperley, D.: Parsing English with a link grammar. In: Proceedings of the Third International Workshop on Parsing Technologies (1993)
12. Lin, D., Pantel, P.: DIRT - Discovery of inference rules from text. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, pp. 323–328 (2001)
13. Akhmatova, E., Molla, D.: Recognizing textual entailment via atomic propositions. In: Quiñonero-Candela, J., Dagan, I., Magnini, B., d'Alché-Buc, F. (eds.) MLCW 2005. LNCS (LNAI), vol. 3944, pp. 385–403. Springer, Heidelberg (2006)
14. Zanzotto, F.M., Moschitti, A.: Automatic learning of textual entailments with cross-pair similarities. In: Proceedings of the 21st COLING and 44th ACL, Sydney, Australia (2006)
15. Zanzotto, F.M., Pennacchiotti, M., Moschitti, A.: A machine learning approach to textual entailment recognition. Natural Language Engineering 15(4), 551–582 (2009)

16. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the Conference on New Methods in Language Processing, Manchester, UK (1994)
17. Lowe, J.B., Baker, C.F., Fillmore, C.J.: A frame-semantic approach to semantic annotation. In: Proceedings of the SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How? (1997)
18. Ruppenhofer, J., Ellsworth, M., Petruck, M.R.L., Johnson, C.R.: FrameNet: Theory and practice (2005)
19. Erk, K., Pado, S.: Shalmaneser – A toolchain for shallow semantic parsing. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy (2006)
20. Tatu, M., Moldovan, D.: A semantic approach to recognizing textual entailment. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP), Vancouver, British Columbia, Canada, pp. 371–378 (2005)
21. Yatbaz, M.A.: RTE4: Normalized dependency tree alignment using unsupervised n-gram word similarity score. In: Proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment, Gaithersburg, USA (2008)
22. Kouylekov, M., Magnini, B.: Recognizing textual entailment with tree edit distance algorithms. In: Proceedings of the First PASCAL Challenges Workshop on Recognizing Textual Entailment, Southampton, UK, pp. 17–20 (2005)
23. Pazienza, M.T., Pennacchiotti, M., Zanzotto, F.M.: Textual entailment as syntactic graph distance: A rule based and a SVM based approach. In: Proceedings of the First PASCAL Challenges Workshop on Recognizing Textual Entailment, Southampton, UK, pp. 25–28 (2005)
24. Clark, P., Harrison, P.: Recognizing textual entailment with logic inference. In: Proceedings of the Fourth PASCAL Challenges Workshop on Recognizing Textual Entailment, Gaithersburg, USA (2008)
25. de Salvo Braz, R., Girju, R., Punyakanok, V., Roth, D., Sammons, M.: Textual entailment recognition based on dependency analysis and WordNet. In: Proceedings of the First PASCAL Challenges Workshop on Recognizing Textual Entailment, Southampton, UK, pp. 29–32 (2005)
26. Bosma, W.E., Callison-Burch, C.: Paraphrase substitution for recognizing textual entailment. In: Peters, C., Clough, P., Gey, F.C., Karlgren, J., Magnini, B., Oard, D.W., de Rijke, M., Stempfhuber, M. (eds.) CLEF 2006. LNCS, vol. 4730, pp. 1–8. Springer, Heidelberg (2006)
27. Burchardt, A., Frank, A.: Approaching Textual Entailment with LFG and FrameNet Frames. In: Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment, Venice, Italy (2006)
28. Ofoghi, B., Yearwood, J.: UB.dmirg: A syntactic lexical system for recognizing textual entailments. In: Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment, Gaithersburg, USA (2009)

A Gradient-Based Metric Learning Algorithm for k-NN Classifiers

Nayyar Abbas Zaidi[1], David McG. Squire[1], and David Suter[2]

[1] Clayton School of Information Technology, Monash University, VIC 3800, Australia,
[2] School of Computer Science, University of Adelaide,
North Terrace SA 5005, Australia
{nayyar.zaidi,david.squire}@infotech.monash.edu.au,
david.suter@adelaide.edu.au

Abstract. The Nearest Neighbor (NN) classification/regression techniques, besides their simplicity, are amongst the most widely applied and well studied techniques for pattern recognition in machine learning. A drawback, however, is the assumption of the availability of a suitable metric to measure distances to the k nearest neighbors. It has been shown that k-NN classifiers with a suitable distance metric can perform better than other, more sophisticated, alternatives such as Support Vector Machines and Gaussian Process classifiers. For this reason, much recent research in k-NN methods has focused on metric learning, i.e. finding an optimized metric. In this paper we propose a simple gradient-based algorithm for metric learning. We discuss in detail the motivations behind metric learning, i.e. error minimization and margin maximization. Our formulation differs from the prevalent techniques in metric learning, where the goal is to maximize the classifier's margin. Instead our proposed technique (MEGM) finds an optimal metric by directly minimizing the mean square error. Our technique not only results in greatly improved k-NN performance, but also performs better than competing metric learning techniques. Promising results are reported on major UCIML databases.

1 Introduction

Nearest neighbor methods for pattern recognition have proven to be very useful in machine learning. Despite their simplicity, their performance is comparable to other sophisticated classification and regression techniques, such as Support Vector Machines (SVM) and Guassian Processes (GP), and they have been applied to a wide variety of problems. For a given query point, a nearest neighbor classifier works by assigning it the label of the majority class in its neighborhood.

It is evident that the k-NN classifier's simplicity is one of its major advantages. A k-NN classifier deals with multi-class classification scenario effortlessly. In contrast, one needs one-versus-one and one-versus-all techniques to deal with multi-class scenarios when using binary classifiers such as SVM. This makes them computationally expensive. As k-NN classifiers need no training, they are computational efficient. Nevertheless, the effectiveness of k-NN methods relies

J. Li (Ed.): AI 2010, LNAI 6464, pp. 194–203, 2010.

on their asymptotic properties. The asymptotic results in [1,2,3] suggest that a 1-NN method based on a simple Euclidean distance will perform well provided the number of training samples is not too small. Indeed 1-NN will approach the performance of a Bayes optimal classifier as the number of training data becomes very large. These asymptotic results are based on the fact that bias in the prediction of function $f(x)$ becomes vanishingly small if the number of training data N is large compared to the number of features p i.e., $N \gg p$. Typical machine learning data, however, has large numbers of features, and the amount of data required to achieve these asymptotic results is unfeasibly large. This is known as the Curse-of-Dimensionality (COD). Another interpretation of the COD is that, in high dimensions, most of the data points are very far apart and k-NN neighborhoods are no longer 'local' [4, section 2.5]. Modifying distances in high dimensions can help to alleviate the COD, reduce bias and make neighborhoods local. This requires a tuned metric—and hence metric learning.

As discussed above, the performance of a nearest neighbor classifier depends critically on two factors: the distance metric used, and size of the neighborhood (specified by k, which denotes the number of nearest neighbors). The value of k controls the Mean Square Error (MSE) which is defined as MSE = bias^2 + variance. Small k implies small bias but high variance, and vice-versa. Since k is specified in terms of the number of nearest neighbors of a query point x, which implicitly depends on a distance measure, MSE can be controlled by estimating a distance metric (a metric is generally specified through a norm and a positive semi-definite matrix). Typically we estimate the inverse square root of the metric. That is, we learn a matrix parameterizing the linear transformation of the input space such that in the transformed space k-NN performs well. If we denote such a transformation by a matrix A, we are effectively learning a metric defined by A^TA such that $d(x, y) = (x - y)^TA^TA(x - y) = (Ax - Ay)^T(Ax - Ay)$.

In the current research on nearest neighbor methods, a dichotomy exists between metric learning methods in terms of their goals. Most 'Metric Learning' algorithms aim to find a metric that results in small intra-class and large inter-class distances [5,6,7,8,9,10]. This results in maximizing the margin.[1]

Figure 1 depicts a simple contrived example of data belonging to two classes represented by red and blue dots. As can be seen, the classes are linearly separable. A hyperplane is indicated by a dark black line. In this scenario, the margin can be maximized in two ways: either we modify the hyper-plane to better fit the training data, or we transform the training data to maximize the margin with respect to a certain hyperplane. The latter has been the goal of most metric learning algorithms. SVMs, on the other hand, optimize the margin by finding an optimal hyperplane. They are designed to minimize empirical risk with a bound on generalization error. Metric learning can also be used to minimize empirical risk i.e., maximize the margin by transforming the training data. Such a strategy has been introduced in [11], where metric learning was introduced as a bias reduction strategy and to reduce MSE to better fit the training data.

[1] The margin of a point is defined as the distance between the point and the closest point on the classification boundary.

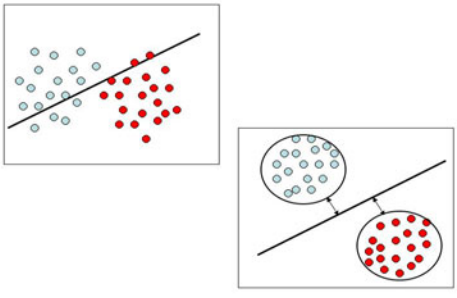


Fig. 1. Contrived example demonstrating the impact of metric on margin

In this paper we present a novel metric learning algorithm with the goals of maximizing the margin by reducing MSE directly. We propose a simple MSE gradient minimization (MEGM - Mean square Error Gradient Minimization) approach to improve the performance of the k-NN neighbor classifier. Our method is based on gradient descent on the MSE objective function. We compare MEGM performance with other metric learning approaches for margin maximization, e.g. neighborhood component analysis (NCA). As shown in section 4, our method not only results in significant improvement in the performance of the k-NN classifier, but also outperforms other metric learning algorithm on most data-sets. As we discuss in section 5, unlike SVM, we minimize the empirical risk only. We do not address generalization in our algorithm, but in our experiments we did not experience any over-fitting. A regularization term can be easily introduced into our framework. This is left as a future work.

The rest of the paper is organized as follows: we discuss related work in section 2. Our proposed MEGM algorithm is described in detail in section 3. A detailed description of our experimental setup and comparitive results on UCIML data-sets are given in section 4. We conclude in section 5 with pointers to future work.

2 Related Work

Our proposed algorithm MEGM is very close in nature to [12] where a gradient based technique is used for selecting relevant features. That is, only diagonal terms of the covariance matrix are estimated. In our method we learn a full covariance matrix rather than estimating only diagonal terms. Thats why MEGM is superior to technique proposed in [12].

The other notable techniques for metric learning are LMNN [13], RCA [14] and NCA [5]. Relevant Component Analysis (RCA) [14] constructs a Mahalanobis distance metric from a weighted sum of in-class covariance matrices. It is similar to Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) in its reliance on second order statistics. Large Margin Nearest Neighbor (LMNN) algorithm in [13] is posed as a convex problem, and thus the

reach of the global solution is guaranteed. However, a special optimization solver is needed for efficient implementation.

Neighborhood Component Analysis (NCA) [5] maximizes margin by minimizing the probability of error under stochastic neighborhood assignment. In particular each point i selects another point j as its neighbor with some probability p_{ij}, and inherits its class labels from the point it selects. p_{ij} is defined as a softmax over Euclidean distances in the transformed space, parameterized by matrix A:

$$p_{ij} = \frac{-\exp(\|Ax_i - Ax_j\|^2)}{\sum_{k \neq i} \exp(-\|Ax_i - Ax_k\|^2)} \tag{1}$$

NCA maximizes the p_{ij} in above equation by finding an optimal A matrix. That is the probability of the number of points correctly classified is maximized. The comparison of our proposed algorithm (MEGM) with NCA has been a major motivation of this work. Though NCA is sound in theory, our empirical results in section 4 suggests that MEGM performs better than NCA on most data-sets. We will mention in section 5 about an approach to combine both MEGM and NCA to improve MEGM's generalization capacity.

3 Approach

In a typical regression setting, an unknown function $f : R^D \rightarrow R$ is predicted from the training data $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ...(\boldsymbol{x}_N, y_N)\}$, where $\boldsymbol{x}_i$ is a data point and y is the corresponding target value. The predicted function $\hat{f}$ is chosen to be the one that minimizes some loss function such as 'mean squared error' (MSE) etc. The MSE for a data set containing N of points is given in the following equation:

$$\text{MSE}(\hat{f}) = \sum_{i=1}^{N} (f(\boldsymbol{x}_i) - \hat{f}(\boldsymbol{x}_i))^2 \tag{2}$$

For classification task having T classes we can replace above error function as:

$$\text{MSE}(\hat{y}) = \sum_{t=1}^{T} \sum_{i=1}^{N} (y_{ti} - \hat{y}_{ti})^2 \tag{3}$$

where $\hat{y}_i$ denotes the predicted probability of point $\boldsymbol{x}_i$ and y_i denotes the true label (either 0 or 1) of point $\boldsymbol{x}_i$. For brevity we have denoted $\hat{y}(\boldsymbol{x_{ti}})$ with $\hat{y}_{ti}$ and $y(\boldsymbol{x_{ti}})$ with y_{ti}. In the following discussion we will assume that there are only two classes to make our derivations simple. For any query point $\boldsymbol{x}_i$, nearest neighbor methods work by predicting the value $\hat{y}_i$ by considering the labels of its k nearest neighbors. In order to have a smooth boundary, each neighbor votes

for the query label based on its distance from the query point (refer to [4] for details). Equation 4 shows the Nadaraya-Watson kernel for regression:

$$\hat{y}(\boldsymbol{x}) = \frac{\sum_j y_j V_j}{\sum_j V_j} \tag{4}$$

The vote V_j casted by each label around the query point $\boldsymbol{x}$ is usually chosen to be a function that decays exponentially as the distance from the query point increases, for example a Gaussian kernel:

$$V_j = \exp\left(\frac{-\mathrm{d}(\boldsymbol{x}, \boldsymbol{x_j})}{2\sigma^2}\right) \tag{5}$$

Determining votes using equation 5 assumes a well defined distance measure. This assumption, as discussed in the previous section, is not always true, due to the COD and irrelevant features, and can lead to bad results. $\mathrm{d}(\boldsymbol{x}, \boldsymbol{x_j})$ in equation 5 can be replaced by a more general metric: that is $\mathrm{d}_L(\boldsymbol{x}, \boldsymbol{x_j})$. If $L = A^T A$, then $\mathrm{d}_L(\boldsymbol{x}, \boldsymbol{x_j}) = (A\boldsymbol{x} - A\boldsymbol{x_j})^T(A\boldsymbol{x} - A\boldsymbol{x_j})$. Since MSE is a function of $\hat{y}$ and $\hat{y}$ depends on $||\boldsymbol{x} - \boldsymbol{x}_j||_L^2$, MSE can be minimized by selecting an optimal value of L. In other words, a change in the L induces a change in the distance, which can alter the votes. This alteration in the votes V_j triggers a change in $\hat{y}$ affecting the MSE. It is more helpful to optimize A rather than L, because optimization for L requires to fulfill semi-positive constraint which is expensive to maintain. Obviously trying all possible values of A is not feasible. Some sort of search mechanism is required to find an optimal value of A. Votes V_j in equation 5 can be replaced by W_j as:

$$W_j = \exp\left(\frac{-\|A\boldsymbol{x} - A\boldsymbol{x_j}\|_2^2}{2\sigma^2}\right) \tag{6}$$

The proposed gradient based technique (MEGM) is based on a gradient descent algorithm to minimize MSE (lets denote by E_A). The gradient E_A is evaluated to find an optimal A matrix. Convergence to the global minimum is not guaranteed. The risk of local minima can be reduced by running the algorithm several times and choosing the output with minimum error E_A. The gradient of E_A with respect to matrix A is:

$$\frac{\partial E}{\partial A} = (y_i - \hat{y}_i)\frac{1}{\sum_j W_j}\sum_j (y_j - \hat{y}_j)\frac{\partial W_j}{\partial A} \tag{7}$$

The size of the Gaussian kernel centered at the query point (σ in equation 6) is set proportional to the distance of the k nearest neighbors. Generally the average distance of half of the nearest neighbors is used, as this measure is more stable under a varying distance metric and in the presence of outliers:

$$\sigma^2 = \frac{1}{2}\frac{1}{P}\sum_{p=1}^{P}\|\boldsymbol{x}-\boldsymbol{x}_p\|^2 \quad \text{where } P = k/2 \tag{8}$$

$\frac{\partial W_j}{\partial A}$ in equation 7 can be derived as:

$$\frac{\partial W_j}{\partial A} = 2W_j A(\boldsymbol{x}-\boldsymbol{x}_j)(\boldsymbol{x}-\boldsymbol{x}_j)^T \tag{9}$$

Combining equations 7 and 9 we can write the gradient of E_A with respect to matrix A as:

$$\frac{\partial E}{\partial A} = 2A(y_i-\hat{y}_i)\frac{1}{\sum_j W_j}\sum_j (y_j-\hat{y}_j)W_j(\boldsymbol{x}-\boldsymbol{x}_j)(\boldsymbol{x}-\boldsymbol{x}_j)^T \tag{10}$$

Equation 10 represents the gradient of the error function with respect to matrix A which is minimized to get an optimal A. The Polack-Ribiere flavour of conjugate gradients is used to compute search directions, and a line search using quadratic and cubic polynomial approximations and the Wolfe-Powell stopping criteria is used together with the slope ratio method for guessing initial step sizes.

4 Experimental Results

In this section we present results on various machine learning databases from UCIML repository [15]. MEGM's results are compared with other metric learning approaches like NCA, RCA and LMNN. The size of neighborhood (k) as discussed in section 3 is consistently set equal to the log2(cardinality of data set) for all databases.

To obtain the final classification results, one nearest neighbor (1-NN) classification is used. As mentioned in section 3, since both NCA and MEGM suffers from local minima problems, some care has to be taken to make sure that it does not effect results. For all databases, we run MEGM and NCA thrice with different training data samples and selected the best results. In order to make sure that our results are not biased to NCA and MEGM due to this procedure, reported results for all other techniques for example k-NN, LMNN and RCA are computed this way. That is each method is run thrice using different training samples and best results are selected in each run. Percentage error rates are reported for all data-sets.

To test the performance of MEGM with other metric learning methods, we selected major UCIML databases. The error rate of each method for different databases is shown in figure 2. The number of data, features and classes for each database is reported in the title. Error rate of each method is obtained using 40 rounds of 2 fold cross-validation. The mean and standard deviation of the results are reported in the performance graphs. Prior to training, all features were normalized to have zero mean and a unit variance.

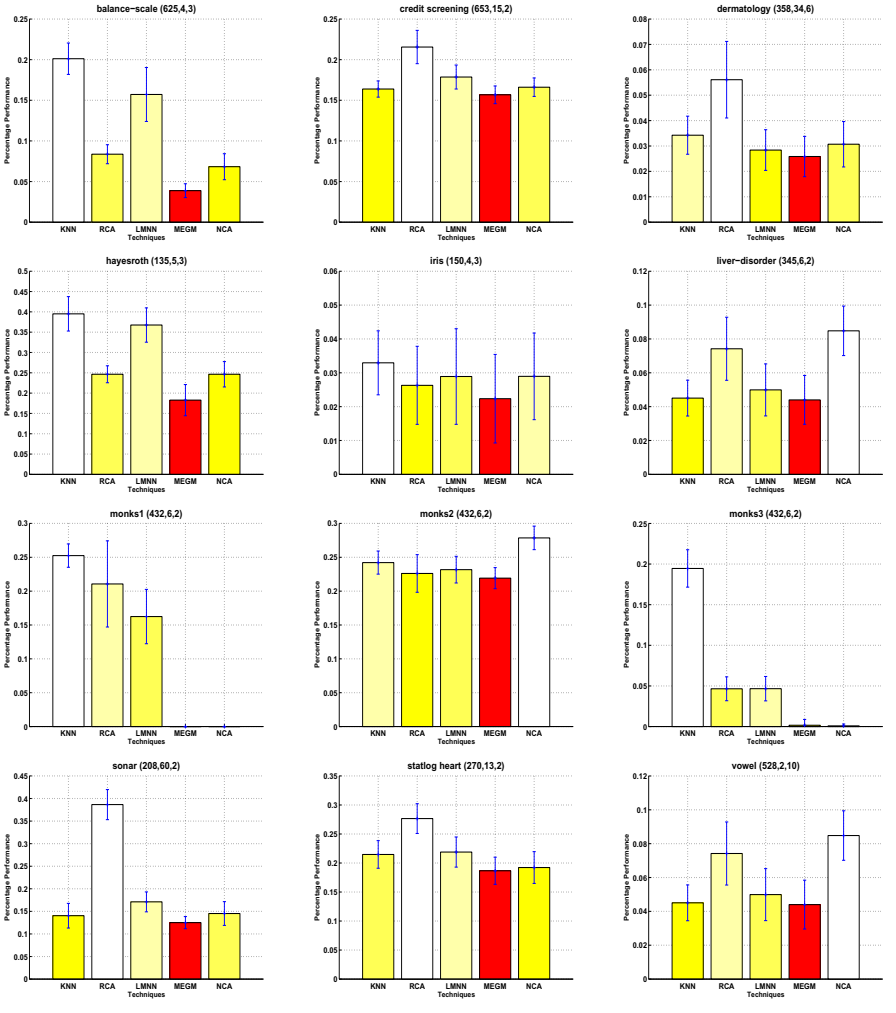


Fig. 2. Error rate comparison of various techniques on UCIML databases

As can be seen MEGM not only improved k-NN classification performance but in most cases resulted in better performance than other metric learning techniques like NCA, RCA and LMNN. MEGM outperforms other methods on Balance and Hayesroth databases. Also it performed marginally better than other techniques on Credit-screeing, Dermatology, Sonar, Statlog-heart, vowel and Monks2. On Monks1 and Monks3 both MEGM and NCA performs equally well and error rate is close to zero for both these methods.

Though MEGM performed better than other approaches on most databases as shown in figure 2, NCA performance is also noteworthy especially on balance, monks1 and statlog-heart. It performed marginally better than other techniques on Ionosphere, Housevote and Hepatitis as shown in figure 3.

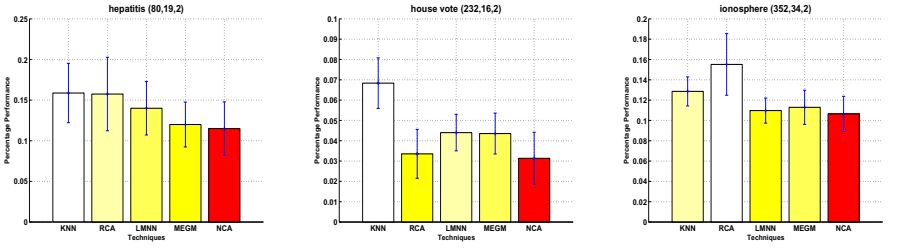


Fig. 3. Error rate comparison of various techniques on UCIML databases, NCA performs best on these data-sets

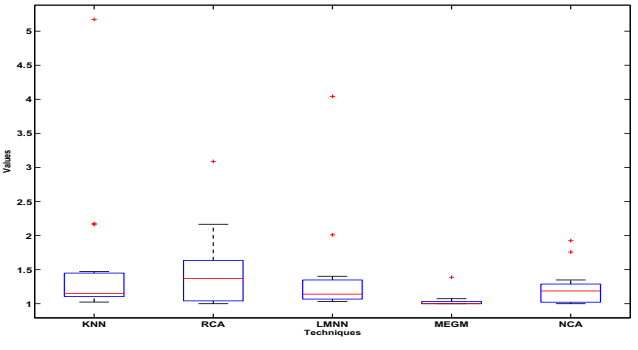


Fig. 4. Box plots depicting the comparison of robustness of different techniques on various UCIML data-sets

To compare the robustness of our algorithm with other algorithms we used the technique described in [11]. This test measures how well a particular method m performs on average in situations that are most favorable to other procedures. Robustness can be measured by computing the ratio b_m of its error rate e_m and the smallest error rate over all other methods that are compared in that example. That is:

$$b_m = \frac{e_m}{\min_{1 \leq k \leq 5} e_k} \tag{11}$$

The best method m^* will have $b_m^* = 1$ and all other methods will have values larger than 1. The larger the value of b_m the worse the performance is of the m^{th} method in relation to the best one for that data-set. Figure 4 shows the distribution of b_m for each method over all 15 UCIML data-sets considered. As can be seen, MEGM turned out to be the most robust of all with NCA coming second. LMNN also performs good except for the presence of outliers.

5 Conclusion

The main pro of our proposed MEGM algorithm is its simplicity. As discussed, MEGM minimizes MSE's gradient using a simple gradient descent algorithm. MEGM improves k-NN classification by learning a data dependent distance metric and performs well on most if not all databases. Also, it deals with multi-class problems effortlessly as opposed to binary classifiers like SVM where a one-versus-one and one-versus-all strategy is used. On the other hand, once a metric is learnt using MEGM, a simple nearest neighbor classification is required. In data-sets where number of classes are very large, nearest neighbor methods should be preferable for their computational efficiency. Therefore k-NN methods equipped with a proper distance metric (for example, one trained with MEGM) can be extremely useful.

A drawback of MEGM includes local minima problem. Standard approaches to avoid local minima are to be used. Also one is tempted to think of over-fitting if the objective function is only MSE. In this work, we did not encounter any over-fitting. As a future work, we are investigating to modify our objective function to include a generalization term, that is penalize large changes in A matrix to avoid over-fitting. We are currently investigating to combine MEGM's and NCA's objective function to improve our results. As in this study, MEGM which is based on the minimization of MSE resulted in better performance than NCA and other metric learning algorithms which maximizes margin explicitly, a natural extension to the proposed method is to combine the two approaches. That is learn a metric by simultaneously maximizing the margin and minimizing the MSE. The objective functions of MEGM and NCA is combined in the following equation:

$$E_A = \sum_{i=1}^{N} \left(y_i - \exp\left(\frac{-\|A\boldsymbol{x} - A\boldsymbol{x_j}\|_2^2}{2\sigma^2} \right) \right) + \left(\frac{\exp(\|Ax_i - Ax_j\|^2)}{\sum_{k \neq i} \exp(-\|Ax_i - Ax_k\|^2)} \right) \quad (12)$$

We are investigating gradient based methods to optimize for A in equation 12. Considering the MEGM results, the combination with NCA can lead to good results.

There has been a lot of work done in adaptive distance metric [16,17]. In adaptive metric learning a separate metric is learnt for each query point. We are currently modifying MEGM to work in such local settings. Training a separate metric for each query point can become computationally expensive. We are investigating clustering techniques to cluster data first and than train a separate metric for each cluster.

In summary, we proposed a simple mean square error's gradient based metric learning algorithm (MEGM) in this paper and showed that MEGM not only results in classification improvement of k-NN classifier but also performs better than other metric learning algorithms. Results are shown on major UCIML databases. Our results are encouraging and requires additional investigation to further improve MEGM performance as described.

References

1. Cover, T.: Rates of convergence for nearest neighbor procedures. In: Proceedings of the International Conference on Systems Sciences (1968)
2. Fix, E., Hodges, J.: Discriminatory analysis - nonparameteric discrimination: consistency properties. Tech. Report, Randolph Field Texas, US Airforce School of Aviation Medicine, Tech. Rep. (1951)
3. Snapp, R., Venkatesh, S.: Asymptotic expansions of the k-nearest neighbor risk. The Annals of Statistics (1998)
4. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics (2001)
5. Goldberger, J., Roweis, S., Hinton, G., Salakhutdinov, R.: Neighborhood component analysis. In: Proceedings of Neural Inforamtion and Processing Systems (2005)
6. Davis, J., Dhillon, I.: Structured metric learning for high dimensional problems. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2008)
7. Weinberger, K., Blitzer, J., Saul, L.: Distance metric learning for large margin nearest neighbor classification. In: Proceedings of Neural Inforamtion and Processing Systems (2005)
8. Sriperumbudar, B., Lang, O., Lanckriet, G.: Metric embedding for kernel classification rules. In: Proceedings of the International Conference on Machine Learning (2008)
9. Globerson, A., Roweis, S.: Metric learning by collapsing classes. In: Proceedings of Neural Inforamtion and Processing Systems (2005)
10. Xing, E., Ng, A., Jordan, M., Russell, S.: Distance metric learning with application to clustering with side-information. In: Proceedings of Neural Inforamtion and Processing Systems (2002)
11. Friedman, J.: Flexible metric nearest neighbor classification. Tech. Report, Dept. of Statistics, Stanford University, Tech. Rep. (1994)
12. Lowe, D.: Similarity metric learning for a variable-kernel classifier. In: Proceedings of Neural Inforamtion and Processing Systems (1996)
13. Weinberger, K., Blitzer, J., Saul, L.: Distance metric learning for large margin nearest neighbor classification. In: Proceedings of Neural Inforamtion and Processing Systems (2006)
14. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning distance functions using equivalence relation. In: Proceedings of the International Conference on Machine Learning (2003)
15. Mertz, C., Murphy, P.: Machine learning repository (2005), http://archive.ics.uci.edu/ml/
16. Hastie, T., Tibshirani, R.: Discriminative adaptive nearest neighbor classification. IEEE Transactions on Pattern Analysis and Machine Intelligence (1996)
17. Zaidi, N., Squire, D.M., Suter, D.: BoostML: An adaptive metric learning for nearest neighbor classification. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (2010)

Dimensionality Reduction for Classification through Visualisation Using L1SNE

Lennon V. Cook and Junbin Gao*

School of Computing and Mathematics,
Charles Sturt University, Bathurst, NSW 2795, Australia
lcook06@postoffice.csu.edu.au, jbgao@csu.edu.au

Abstract. Dimensionality Reduction algorithms have wide precedent for use in preprocessing for classification problems. This paper presents a new algorithm, based on a modification to Stochastic Neighbour Embedding and t-Distributed SNE to use the Laplacian distribution instead of, respectively, the Gaussian Distribution and a mismatched pair of the Gaussian Distribution and Student's t-Distribution. Experimental results are presented to demonstrate that this modification yields improvement.

1 Introduction

Recent years have seen a large increase in studies on dimensionality reduction (DR) algorithms, the goal of which is to find a lower-dimensional representation of high-dimensional data [20]. A common motive for this is to overcome the "curse of dimensionality" [20,6] that the complexity of many algorithms is bound by the dimension of the data, and can become intractable in many real-world datasets where the dimensionality is quite high. Application areas include manifold learning[11], pattern recognition [13], data mining[15], data classification[2] and data visualisation [19,8].

Originally, the focus was on algorithms such as Principal Component Analysis (PCA) [10], and Linear Discriminant Analysis (LDA) [4] which assume linearity - that is, that the low-dimensional vectors are a linear combination of the original high-dimensional vectors. More recently, however, much improvement has been gained by relaxing this constraint. Resulting Non-Linear Dimensionality Reduction (NLDR) algorithms include, for example, Local Linear Embedding (LLE) [16], Lapacian Eigenmaps (LE) [1], Isometric mapping (Isomap) [17], Local Tangent Space Alignment (LTSA) [21], and Gaussian Process Latent Variable Model (GPLVM) [12].

The Stochastic Neighbour Embedding (SNE) algorithm is one of these non-linear dimensionality reduction techniques [7], which considers the probability that any two points will be neighbours. A major justification of this technique is

* Corresponding author.

J. Li (Ed.): AI 2010, LNAI 6464, pp. 204–212, 2010.

that it can be readily extended to more complex relationships between the high- and low-dimensional data than the strictly one-to-one mapping of many earlier algorithms. This is presented as being useful in classification of documents based on the words they contain, where a strict one-to-one relationship fails to account for a single word having multiple meanings. However, SNE suffers the so-called "crowding problem" [19], where distances between points in the high-dimensional space are reflected less accurately in the low-dimensional space as they increase. To counteract this, the authors of [19] propose that mismatched distributions for the neighbouring probabilities be used, suggesting a heavier-tailed distribution for the lower-dimensional vectors. They choose the Student-t distribution for this, and their experiments demonstrate the new algorithm, tSNE, outperforms most existing nonlinear dimensionality reduction algorithms.

Using the Student-t distribution, which is a heavy-tailed generalisation of the Gaussian distribution used by SNE, increases the robustness of the model against the crowding problem. The same goal can be achieved by instead using the centered Laplacian distribution (or L1 distribution or the least absolute deviance). The L1 distribution is much less sensitive to outliers compared to the Gaussian density and also has only one tunable parameter, while the Student-t distribution is determined by two parameters (the degrees of freedom and the scale parameter). The approach of using the L1 distribution originates from LASSO [18], and has caught some interest in machine learning [14] and statistics. Besides the robustness against outliers context, L1 distribution assumption is also used as a penalty/regularization term on model parameters to enforce sparsity, or parameter/feature selection, such as sparse PCA [9,22], and logistic regression [14]. A recent paper [3] gives a detailed analysis on the generalized distribution which includes L1 as a special case.

This paper presents a new dimensionality reduction algorithm for data visualisation to aid classification based on the generalized L1 distribution in the classical SNE. This new algorithm, similarly to tSNE, differs in the optimisation problem which is solved - that is, this paper presents the case that this technique provides a better match between the formal optmisation problem and the abstract research goal, rather than presenting a better way of finding a solution to the same optimisation problem.

The paper is organised as follows. Section 2 presents a revision of the existing background work, focusing on two algorithms for Dimensionality Reduction. Section 3 presents the proposed new algorithm. Following this, experiments for classification on several datasets are conducted in Section 4 and results are analysed and compared against the results of the existing algorithms. Finally, the conclusions drawn from these experiments are presented.

For the remainder of the paper, $X = \{\boldsymbol{x}_i\}_{i=1}^N$ is the set of D-dimensional input vectors, and $Y = \{\boldsymbol{y}_i\}_{i=1}^N$ is the set of d-dimensional reduced vectors. It can be assumed from the direct goal of dimensionality reduction that $d \ll D$.

2 SNE and *t*-SNE

2.1 Formulation of SNE

The algorithm works by determining which pairs of points should be matched in terms of their Euclidean distance in the high- and low-dimensional spaces, respectively. The matching is based on the normalized discrete distribution determined by their Gaussian kernels in the two spaces. For the high-dimensional input space, the probability that any x_i and x_j are neighboured is:

$$p_{ij} = \frac{\exp\left(-\frac{1}{\sigma_i^2}\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2\right)}{\sum_{k \neq i} \exp\left(-\frac{1}{\sigma_k^2}\left\|\boldsymbol{x}_i - \boldsymbol{x}_k\right\|^2\right)} \tag{2.1}$$

where $\|\boldsymbol{x}\|$ denotes the Euclidean norm of $\boldsymbol{x}$. $P_i = \{p_{ij} \forall j\}$ is the Gaussian-distributed set of all such probabilities for a particular $\boldsymbol{x}_i$. The low-dimensional neighbouring probabilities are calculated the same way:

$$q_{ij} = \frac{\exp\left(-\left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|^2\right)}{\sum_{k \neq i} \exp\left(-\left\|\boldsymbol{y}_i - \boldsymbol{y}_k\right\|^2\right)} \tag{2.2}$$

And likewise $Q_i = \{q_{ij} \ \forall j\}$.

SNE aims to select $\boldsymbol{y}_i$ so that each Q_i matches its associated P_i as closely as possible. This leads to a cost function of:

$$C = \sum_i KL\left(P_i||Q_i\right) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{2.3}$$

Where $KL\left(P_i||Q_i\right)$ is the Kullback-Leibler divergence between the distributions.

2.2 *t*-Distributed SNE

The t-distributed SNE algorithm is originally presented by [19]. It is formulated as a modification to SNE in which the Gaussian distribution (2.1) is retained in the high-dimensional space, while the heavier-tailed Student-t distribution is used for the low-dimensional space. This leaves the calculation for each p_{ij} identical to the calculations in SNE. However, the calculations for q_{ij} change to:

$$q_{ij} = \frac{\left(1 + \left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|^2\right)^{-1}}{\sum_{k \neq i} \left(1 + \left\|\boldsymbol{y}_k - \boldsymbol{y}_i\right\|^2\right)^{-1}} \tag{2.4}$$

In this formulation(2.4), as in [19], the degress of freedom parameter in the Student-t distribution is assumed to be 1. The cost function, like SNE, is given from Kullback-Leibler divergences (2.3).

3 Laplacian-distrubted (L1) SNE

SNE's P_i and Q_i distributions measure the probability of points being neighboured. The difference between them is not that they measure this on different data, but rather that they measure it on different representations of the data. Therefore, it makes sense that the particularly probability distributions should be identical.

This is at odds with tSNE's use of a different distribution for each Q_i than for its corresponding P_i, and suggests that the crowding problem it seeks to solve is caused by an inappropriate choice of distribution in the first place. The choice of a Gaussian distribution in SNE is never justified by [7], but simply taken as a default. [5] notes that this is a common default choice, but suggests that it is not necessarily justified by statistics theory.

It seems sensible, then, that the argument presented in [5] can apply here, and that hence the Laplacian (L1) distribution may be a better choice than the Gaussian for the neighbourhood probability model of SNE.

Therefore, we define:

$$p_{i|j} = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_1}{2\sigma_i^2}\right) \tag{3.1}$$

where $|\boldsymbol{x}|_1 = |x_1| + |x_2| + |x_3| + ...|x_D|$ is the L1-norm of $\boldsymbol{x}$, and σ_i^2 is the variance of $\boldsymbol{x}_i$. In order to constrain symmetry in $p_{ij} = p_{ji}$ and hence simplify the cost function, we then define:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n} \tag{3.2}$$

with $p_{ii} = 0$.

And similarly in the low dimensional space:

$$q_{ij} = \frac{\exp\left(-\left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|_1\right)}{\sum_{k \neq l} \exp\left(-\left\|\boldsymbol{y}_k - \boldsymbol{y}_l\right\|_1\right)} \tag{3.3}$$

with $q_{ii} = 0$.

Again, we seek to minimise the cost function (2.3) given by the Kullback-Leibler divergences between P_i and Q_i . Like the authors of [19], we do this by iterative gradient descent. Since the only distances $d_{ij} = \left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|_1$ affected by a change in any particular y_i are d_{ij} and d_{ji} for all j, we have:

$$\frac{\partial C}{\partial y_i} = 2 \sum_j \frac{\partial C}{\partial d_{ij}} \operatorname{sgn}\left(\boldsymbol{y}_i - \boldsymbol{y}_j\right)$$

where $\operatorname{sgn}(\boldsymbol{y}) = \left(\frac{y_1}{|y_1|}, \frac{y_2}{|y_2|}, \ldots, \frac{y_d}{|y_d|}\right)$ is the sign vector of $\boldsymbol{y}$.

From 2.3, we can find:

$$\begin{aligned}\frac{\partial C}{\partial d_{ij}} &= \sum_{k \neq l} p_{kl} \frac{\partial \log q_{kl}}{\partial d_{ij}} \\ &= p_{ij} - q_{ij} \sum_{k \neq l} p_{kl} \\ &= p_{ij} - q_{ij} \\ \therefore \frac{\partial C}{\partial y_i} &= 2 \sum_j (p_{ij} - q_{ij}) \operatorname{sgn}\left(\boldsymbol{y}_i - \boldsymbol{y}_j\right) \end{aligned} \tag{3.4}$$

This form has roughly equivalent computationally complexity to the gradient of tSNE. Also, as mentioned in the introduction, it lacks the degrees of freedom parameter which tSNE requires to be tuned to the particular dataset.

4 Experimental Results

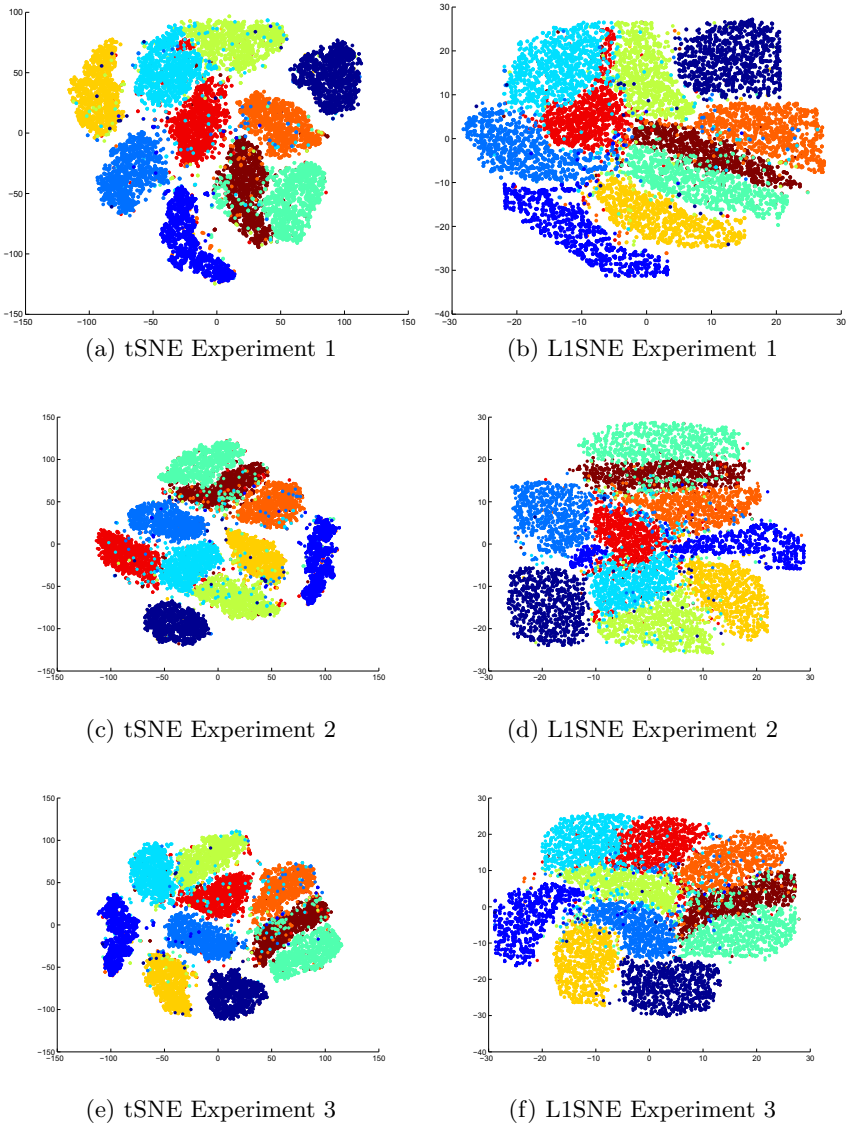
Experiments were run to compare tSNE and L1SNE on three datasets with class information, each of which is detailed in the following subsections. Both algorithms are optimised using the iterative gradient descent method mentioned previously. In all cases, the initial solution is given randomly, and the optimisation is terminated after a fixed number of iterations. The implementation of tSNE is from Laurens van der Maaten's MATLAB Toolbox for Dimensionality Reduction (`http://ict.ewi.tudelft.nl/~lvandermaaten/Matlab_Toolbox_for_Dimensionality_Reduction.html`)

For each experiment, graphs are presented to show the results. These are colourised to show the true class information of each point, which is not made available to the algorithms as they run. Quantitative errors are also provided for each experiment, calculated as a modidfied KNN-error called the k-point local clustering error (KLCE). This calculation alleviates KNN's need for a backprojection for new high-dimensional points to be mapped individually onto a precalculated low-dimensional space. KLCE considers each low-dimensional point in turn, and deduces what proportion of the k nearest points are of a different true class to the point under consideration. The final error is the average of these across the reduced dataset.

4.1 Handwritten Digits

Images of handwritten digits 0-9 were selected randomly from the MNIST dataset, with 1,500 images taken per digit. The results are presented on a two-dimensional scatterplot, colourised to reflect the digit each point represents. Results are provided for three different random subsets to demonstrate repeatability.

It can be seen in Figure 1 that both L1SNE and tSNE successfully group the data into tight clusters in the lower dimensional space according to the digit they represent. L1SNE does not spread the distinct clusters as clearly from



(a) tSNE Experiment 1

(b) L1SNE Experiment 1

(c) tSNE Experiment 2

(d) L1SNE Experiment 2

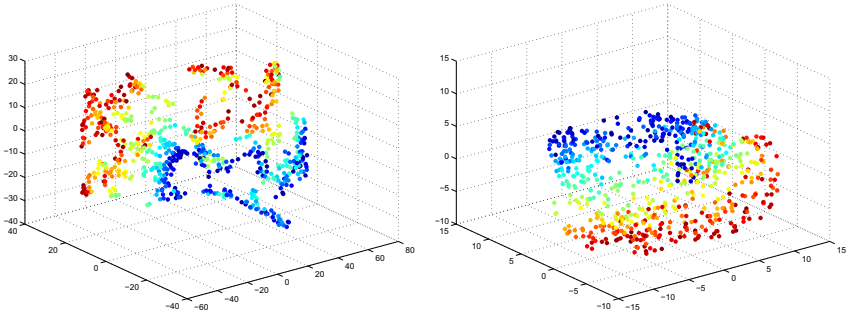
(e) tSNE Experiment 3

(f) L1SNE Experiment 3

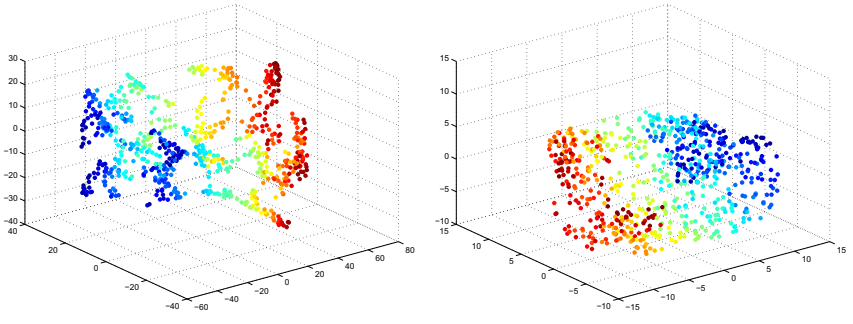
Fig. 1. Handwritten Digits Experiments

Table 1. KLCE Errors for Handwritten Digits Experiments

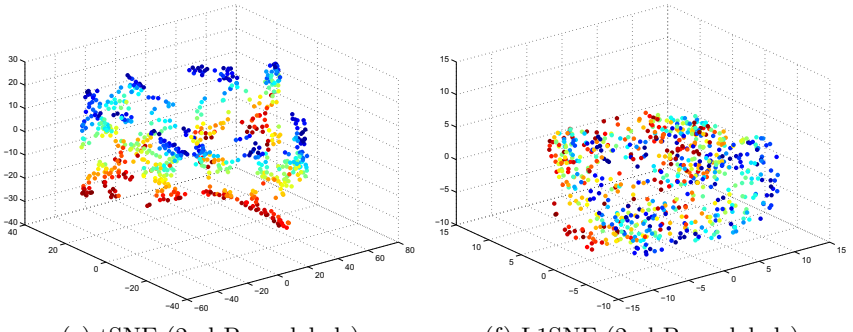
Experiment	1	2	3
tSNE	0.0723	0.0731	0.0847
L1SNE	0.0899	0.0937	0.1043



(a) tSNE (Lights labels) (b) L1SNE (Lights labels)



(c) tSNE (1st Poses labels) (d) L1SNE (1st Poses labels)



(e) tSNE (2nd Poses labels) (f) L1SNE (2nd Poses labels)

Fig. 2. Photographs of Faces

one another as does tSNE, however it shows less tendency for small groups of data to 'wander' between clusters. In the cases where it does do this, the offending group tends to appear far nearer to its proper cluster, as well as visibly smaller in number, than in tSNE. L1SNE's tendancy to place the clusters much closer together in the space appears to be cause of its slightly higher error rate (Table 1): points near to the edge of have many close points which belong to a neighbouring cluster. If the clusters were spread further from each other, as in

tSNE, then points deeper in the same cluster would be closer than points on the nearest edge of a nearby cluster.

4.2 Photographs of Faces

This experiment is designed to show utility in reducing to higher than three dimensions, as well as when multiple, independent, classification schemes are available for the data. The data is taken from the Faces dataset, which contains photgraphs of faces in varying poses and with changed lighting. Each datapoint is the pixel data from a 64-by-64 pixel image, giving 698 4096-dimensional points. This data set comes with three sets of labels, two representing the pose of the subject and one representing the lighting of the photograph. Three experiments were performed to test the utility of reduction to different low-dimensional spaces. Data are reduced by each algorithm to four dimensions. Since the labelling information for this dataset is continuous rather than discrete, KLCE errors are unavailable for this experiment. Instead, the data is projected onto a 3-dimensional space using Principle Component Analysis[10] so that it can be graphed for visual inspection.

The results in Figure 2 suggest that classification by L1SNE would be more accurate on two of the three sets of labels. Once again, it can be seen that L1SNE's primary drawback is a failure to clearly demarkate the boundaries of each class of points, but that it shows improvement on the existing algorithm other than this.

5 Conclusions

In these experiments, L1SNE shows improvement over its predecessors. It shows particular strength in the case where the reduced dimension is greater than three, and where there are multiple, independent, ways to classify the data. The primary identifiable issue which could interfere with automatic classification of data is that it fails to separate the classes of points sufficiently far from each other, so that points on the very edge of a class cluster may appear to belong to the neighbouring cluster. This could be overcome, for example, by partially human-assisted classification in ambiguous edge cases.

The gradient descent method used here for optimisation is rather trivial, as are the random initial solution and the fixed-iterations termination condition. Future research may look for improved techniques in all of these areas, and this may do much to improve the results of all of the algorithms tested here.

However, since these issues were identical across both algorithms, both algorithms could be reasonably expected to improve by equal margins. The experiments conducted argue that the L1SNE cost function has a minimum which is closer to the abstract "ideal" solution, regardless of the optimisation method used to search for it. The evidence then suggests that, of these two, L1SNE would continue to give the better results under such improvement, while maintaining relatively low computational complexity compared to the original SNE algorithm.

References

1. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15(6), 1373–1396 (2003)
2. Buchala, S., Davey, N., Frank, R.J., Gale, T.M.: Dimensionality reduction of face images for gender classifcation (2004)
3. Caron, F., Doucet, A.: Sparse bayesian nonparametric regression. pp. 88–95 (2008)
4. Fisher, R.A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics 7, 179–188 (1936)
5. Gao, J.: Robust L1 principal component analysis and its Bayesian variational inference. Neural Computation 20(2), 555–572 (2008)
6. Guo, Y., Kwan, P.W.H., Hou, K.X.: Visualization of protein structure relationships using constrained twin kernel embedding (2008)
7. Hinton, G., Roweis, S.: Stochastic neighbour embedding. In: Roweis, S. (ed.) Advances in Neural Information Processing Systems, vol. (15), pp. 833–840. MIT Press, Cambridge (2003)
8. Huang, S., Ward, M., Rundensteiner, E.: Exploration of dimensionality reduction for text visualisation. In: Proceedings of the Third Internations Conference on Coordinated and Multiple Views in Exploratory Visualisation (2005)
9. Jolliffe, I.: Principal component analysis, 2nd edn. Springer, New York (2002)
10. Jolliffe, M.: Principal Component Analysis. Springer, New York (1986)
11. Kentsis, A., Gindin, T., Mezei, M., Osman, R.: Calculation of the free energy and cooperativity of protein folding (May 2007)
12. Lawrence, N.: Probabilistic non-linear principal component analysis with gaussian process latent variable models. Journal of Machine Learning Research 6, 1783–1816 (2005)
13. Lima, A., Zen, H., Nankaku, Y., Tokuda, K., Miyajima, C., Kitamura, T.: On the use of kernel pca for feature extraction in speech recognition (2004)
14. Ng, A.: Feature selection, L1 vs. L2 regularization, and rotational invariance. In: Proceedings of Intl. Conf. Machine Learning (2004)
15. Oliveria, S., Zaïane, O.: Privacy-preserving clustering by object similarity-based representation and dimensionality reduction transformation. In: Workshop on privacy and security aspects of data mining, pp. 21–30 (2004)
16. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290(22), 2323–2326 (2000)
17. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290(22), 2319–2323 (2000)
18. Tibshirani, R.: Regression shrinkage and selection via the LASSO. J. Royal. Statist. Soc B. 58, 267–288 (1996)
19. van der Maaten, L., Hinton, G.: Visualising data using t-sne (2008)
20. van der Maaten, L., Postma, E.O., van den Hick, H.J.: Dimensionality reduction: A comparative review (2008)
21. Zhang, Z., Zha, H.: Principal manifolds and nonlinear dimensionality reduction via tangent space. SIAM Journal on Scientific Computing 26(1), 313–338 (2005)
22. Zou, H., Hastie, T., Tibshirani, R.: Sparse principal component analysis. Technical report, Statistics Department, Stanford University (2004)

Review of Modern Logistic Regression Methods with Application to Small and Medium Sample Size Problems

Enes Makalic and Daniel Francis Schmidt

The University of Melbourne, Centre for MEGA Epidemiology,
723 Swanston Street, Carlton VIC 3053, Australia
{emakalic,dschmidt}@unimelb.edu.au

Abstract. Logistic regression is one of the most widely applied machine learning tools in binary classification problems. Traditionally, inference of logistic models has focused on stepwise regression procedures which determine the predictor variables to be included in the model. Techniques that modify the log-likelihood by adding a continuous penalty function of the parameters have recently been used when inferring logistic models with a large number of predictor variables. This paper compares and contrasts three popular penalized logistic regression methods: ridge regression, the Least Absolute Shrinkage and Selection Operator (LASSO) and the elastic net. The methods are compared in terms of prediction accuracy using simulated data as well as real data sets.

Keywords: Logistic regression; Variable Selection; LASSO; Elastic Net; Ridge regression.

1 Introduction

Logistic regression is one of the most widely applied machine learning algorithms in binary classification problems. The popularity of logistic regression is due to the relatively low computational complexity of fitting the model parameters and the high interpretability of the logistic model. Fitting a logistic regression model is commonly done using the iteratively-reweighted least squares (IRLS) algorithm which is efficient provided the number of predictor variables is not too large. Most statistical software packages in use today implement some form of IRLS; for example, STATA© has the logit() function. Unlike black-box algorithms, such as artificial neural networks, logistic regression models are highly interpretable and thus can be used in scenarios where interpretation of the model is as important as prediction accuracy.

Formally, one observes p predictor variables (or covariates) $\mathbf{x} \in \mathbb{R}^p$ and a binary target (or response) variable $y \in \{-1, +1\}$ which states the class membership of the observed vector of predictors. The conditional probability that a vector of covariates $\mathbf{x}$ is assigned to a class y in logistic regression is

$$p(y = \pm 1 | \mathbf{x}, \boldsymbol{\beta}) = \frac{1}{1 + \exp(-y\mathbf{x}'\boldsymbol{\beta})} \tag{1}$$

J. Li (Ed.): AI 2010, LNAI 6464, pp. 213–222, 2010.

where $\boldsymbol{\beta} \in \mathbb{R}^p$ are the regression coefficients. A positive regression coefficient for a predictor implies that the predictor is associated with an increased probability of the response ($y = +1$), while a negative coefficient reduces the response probability. A predictor with a regression coefficient of zero has no effect on the probability of the response and should ideally be excluded from the model.

In order to make accurate predictions, one must estimate the parameter coefficients and select which of the p predictors, if any, are useful in explaining the response. In practice, one typically observes a set of independent observations $\mathrm{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ and proceeds by forming the log-likelihood

$$l(\boldsymbol{\beta}) = -\sum_{i=1}^{n} \log\left(1 + \exp(-y_i \mathbf{x}_i' \boldsymbol{\beta})\right) \tag{2}$$

which is a function of the regression parameters $\boldsymbol{\beta}$. A commonly used estimator of $\boldsymbol{\beta}$ is the the maximum likelihood estimator obtained by finding the parameters $\boldsymbol{\beta}$ that maximise the log-likelihood function.

Traditionally, a stepwise regression procedure combined with the maximum likelihood method for parameter inference is used to determine significant predictors and their contribution to the probability of the target variable. The decision to include a new regressor in the model at each stage in the selection process is then based on the one set of data. This practice of data re-use, or data dredging, is known to be problematic and can often result in biased selection of significant predictors.

More recently, techniques that modify the log-likelihood by adding a penalty function of the parameters have been used when inferring logistic models with a large number of predictor variables. This paper considers three popular penalized logistic regression algorithms: ridge regression [6], the Least Absolute Shrinkage and Selection Operator (LASSO) [11] and the elastic net [12]. Analysis of these methods in the literature has largely focused on the linear regression model in the asymptotic setting (that is, as the sample size or the number of predictors approaches infinity). In contrast, there has been relatively little research on the performance of penalized logistic regression methods, especially in small and medium sample size problems. This is somewhat surprising given the popularity of logistic regression in practice.

This paper aims to address the aforementioned gap by comparing and contrasting three popular logistic regression algorithms in the small to medium sample size setting. Due to the mathematical complexity of the logistic model, all performance comparisons will be empirical and use both simulated data as well as real data sets.

2 Logistic Regression Algorithms

2.1 Stepwise Regression

Stepwise regression encompasses a range of procedures for automatic selection of significant predictors for linear or logistic regression models. Stepwise regression

algorithms can be divided into two broad categories: (1) forward selection, and (2) backward elimination. In forward selection, the algorithm starts with all predictors excluded from the model and, at each step, includes a single predictor variable that is statistically important. For example, statisticians often include a predictor that results in the largest reduction of the log-likelihood at each stage in the selection process. The process is continued until all variables are included in the model or until some pre-determined threshold is met. In contrast, backward elimination begins with the full model that includes all p predictors and at each step removes a predictor from the model that does not result in a significant change to the log-likelihood. Like in forward selection, the elimination process continues until either all predictors are excluded or some performance threshold is met. It is also possible to use a hybrid strategy that considers inclusion and removal of regressors at each step. Due to the similarity of the various stepwise regression procedures, this paper considers only the forward selection algorithm.

Given a model proposed by forward selection, it remains to infer the model parameters $\boldsymbol{\beta}$. The most popular approach of parameter inference and the one considered in this paper is that of maximum likelihood. The maximum likelihood estimator is the estimator that maximises the likelihood or the log-likelihood (2). Formally, the maximum likelihood estimator of the logistic model parameters is

$$\hat{\boldsymbol{\beta}}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\beta}} \left\{ -\sum_{i=1}^{n} \log\left(1 + \exp(-y_i \mathbf{x}_i' \boldsymbol{\beta})\right) \right\} \tag{3}$$

The maximum likelihood estimator must be obtained numerically. Due to the convexity of the log-likelihood, the estimator is unique unless the data is completely (linearly) separable.

The main disadvantage of forward selection and stepwise regression in general is that the same data set is used for testing whether a predictor is included (or excluded) at each stage in the algorithm. Each test is therefore conditioned on the assumption that the previous inclusion (or exclusion) of predictors is 'correct'. Statistical corrections for this multiple testing problem are possible, but are by no means trivial. Consequently, forward selection can result in a biased selection of pertinent predictors.

3 Penalized Logistic Regression

It is well known that the method of maximum likelihood often overestimates logistic regression parameters resulting in models that predict poorly. In order to tackle this problem a number of methods that shrink the parameters and perform automatic variable selection have been proposed. This paper examines the three most popular penalized logistic regression methods in use today: (1) ridge regression, (2) the Least Absolute Shrinkage and Selection Operator (LASSO) [11], and (3) the elastic net [12]. Note that forward selection may also be viewed as a type of penalized regression where the penalty function is the ℓ_0 norm. A Bayesian implementation of penalized logistic regression is briefly considered in Section 4.

3.1 Ridge Regression

Ridge regression [6,2] maximises the log-likelihood subject to a penalty on the size of the regression parameters. This results in continuous shrinking of the maximum likelihood parameter estimates which is known to improve prediction accuracy. Formally, the ridge regression estimator is

$$\hat{\boldsymbol{\beta}}_{\mathrm{RR}} = \arg\max_{\boldsymbol{\beta}} \left\{ l(\boldsymbol{\beta}) - \lambda \sum_{i=i}^{p} \beta_i^2 \right\} \tag{4}$$

where the penalty parameter $\lambda \geq 0$ determines the amount of shrinkage; for example, $\lambda = 0$ denotes no shrinkage, while $\lambda \to \infty$ results in all parameters being shrunk to zero. Ridge regression can often result in models that predict well but cannot infer sparse models, a sparse model being one where majority of the coefficients are zero. Ridge regression either shrinks all parameters to zero or includes all predictors in the model.

3.2 Least Absolute Shrinkage and Selection Operator

Least absolute shrinkage and selection operator (LASSO) [11] employs a ℓ_1 penalty on the regression parameters rather than the ℓ_2 penalty of ridge regression. This results in a procedure that simultaneously performs parameter shrinkage and pertinent variable selection automatically. Formally, the LASSO estimator is

$$\hat{\boldsymbol{\beta}}_{\mathrm{LA}} = \arg\max_{\boldsymbol{\beta}} \left\{ l(\boldsymbol{\beta}) - \lambda \sum_{i=i}^{p} |\beta_i| \right\} \tag{5}$$

where $\lambda \geq 0$ determines the amount of shrinkage. The main advantage of LASSO over ridge regression is that it can generate sparse models which are easier to interpret. However, the LASSO can exhibit undesirable behaviour in the presence of highly correlated predictors. For example, if there is a group of predictors which are highly correlated, the LASSO tends to randomly include one predictor from the group ignoring the other predictors. In addition, empirical observations [11] show that ridge regression dominates the LASSO in linear models when there are many correlated predictors.

3.3 Elastic Net

Recently, Zou and Hastie have proposed a new penalized regression technique, the elastic net [12], to address the poor performance of LASSO given many correlated predictors. Like the LASSO, elastic net can produce sparse models as well as shrink parameter coefficients. However, if there is a group of highly correlated predictors, the elastic net includes all the predictors in the group thereby improving prediction performance. The elastic net penalty is a combination of the LASSO and ridge regression penalties, defined as

$$\hat{\boldsymbol{\beta}}_{\mathrm{EN}} = \arg\max_{\boldsymbol{\beta}} \left\{ l(\boldsymbol{\beta}) - \lambda_1 \sum_{i=i}^{p} |\beta_i| - \lambda_2 \sum_{i=i}^{p} \beta_i^2 \right\} \tag{6}$$

where $(\lambda_1, \lambda_2) \geq 0$ are the penalty parameters. In contrast to ridge regression and the LASSO, elastic net requires inference of two extra penalty parameters which can be computationally demanding. Furthermore, it has been observed that the elastic net can overshrink the regression parameters in linear models. An empirical correction for this behaviour is available but only for the linear regression model.

4 Bayesian Logistic Regression

Penalized regression methods can be interpreted within the Bayesian statistics paradigm. Here, the log-likelihood penalty takes the form of a prior distribution over the parameter coefficients $\boldsymbol{\beta}$. For example, ridge regression can be viewed as a Bayesian method where the prior for the parameters is a multivariate Gaussian distribution; the variance of the distribution being a function of the penalty parameter λ. Similarly, the LASSO penalty amounts to assuming a Laplace prior distribution over the parameter coefficients. Inference proceeds by forming a posterior distribution of the parameters conditioned on the data.

Bayesian algorithms for penalized logistic regression have received relatively little attention, perhaps due to the mathematical complexity of the logistic model. Bayesian sampling approaches based on various approximations to the posterior distribution are examined in [9,10]. An exact sampling scheme for the logistic distribution through data augmentation was introduced in [7]. Recently, an efficient scheme based on z-distributions is given by Gramacy and Polson [4]. We briefly examine how the Bayesian interpretation of the various penalized regression methods compares to the standard algorithms in Section 5.2.

5 Empirical Comparison

Although a number of papers review penalized logistic regression [1,5], relatively little work has been published on empirical performance of modern logistic regression methods in small to medium sample size applications. The methods chosen here represent the most popular approaches to penalized logistic regression in the literature. The all-subset selection algorithm was not tested because of the inherent instability (high variance) as well as prohibitive computational complexity of the method; for p regressors, all-subset selection requires fitting 2^p logistic models! Various extensions of the ridge regression and LASSO, such as for example group LASSO, which have been developed for correlated predictors have also been omitted. A number of these algorithms require that the correlation groups are pre-defined before executing the method. This is generally not possible in practice. Lastly, the DANTZIG [8] selector was not included in the simulation experiments due to it's high similarity to the LASSO.

The selected logistic regression methods are now compared using both real and simulated data. All simulation code was written in the MATLAB© environment and is available from the authors upon request. Special care was taken during implementation of numerical optimisation routines since both the LASSO

and elastic net are not differentiable at $\beta_i = 0$. Furthermore, standard Newton-Raphson optimisation requires costly matrix inversion and is only feasible when the number of predictors is relatively small. We have implemented a cyclic coordinate descent algorithm [3] which can be applied to a large number of predictors and gracefully handles minimisation of the LASSO and elastic net functions.

5.1 A Simulation Study

We have simulated data sets from four different scenarios originally examined in [12] and reproduced here for convenience. For each simulation, we independently generated a training set, a validation set and a test set. All regression parameters were estimated using only the training data. The validation data set was used to select the optimal penalty parameters, for the ridge, LASSO and elastic net algorithms, and the best model for the forward selection procedure. The optimal penalty parameters were computed using a grid search over a lattice recommended in [12]. The performance of the methods was computed using only the test data set. We have chosen the entropy loss function, rather than 0/1 loss, for this purpose. Unlike the 0/1 loss, the entropy loss is a convex, differentiable function that uses estimated probabilities to compute classification error and does not require specification of an arbitrary hard threshold. The entire procedure was repeated for 1000 iterations. The notation $\cdot/\cdot/\cdot$ is subsequently used to denote the sample size of the training, validation and test sets respectively. The four scenarios are:

1. Example 1: The number of samples was 20/20/200. The true regression coefficients were set to $\boldsymbol{\beta} = (3, 1{\cdot}5, 0, 0, 2, 0, 0, 0)'$. The pairwise correlation between predictors i and j was $\text{corr}(i, j) = 0.5^{|i-j|}$.
2. Example 2: Same as Example 1, except that $\beta_i = 0.85$ for all i.
3. Example 3: The number of samples was 100/100/400 and $\text{corr}(i, j) = 0.5$ for all i and j. The regression coefficients ($p = 40$) were
$$\boldsymbol{\beta} = (\underbrace{0, \ldots, 0}_{10}, \underbrace{2, \ldots, 2}_{10}, \underbrace{0, \ldots, 0}_{10}, \underbrace{2, \ldots, 2}_{10})'$$
4. Example 4: The number of samples was 50/50/400. The regression coefficients ($p = 40$) were
$$\boldsymbol{\beta} = (\underbrace{3, \ldots, 3}_{10}, \underbrace{2, \ldots, 2}_{15}, \underbrace{0, \ldots, 0}_{25})'$$
The predictors were generated from:
$$\begin{aligned} \mathbf{x}_i &= Z_1 + \epsilon_i, \quad Z_1 \sim N(0, 1), \quad i = 1, \ldots, 5 \\ \mathbf{x}_i &= Z_2 + \epsilon_i, \quad Z_2 \sim N(0, 1), \quad i = 6, \ldots, 10 \\ \mathbf{x}_i &= Z_3 + \epsilon_i, \quad Z_3 \sim N(0, 1), \quad i = 11, \ldots, 15 \end{aligned}$$
and $\mathbf{x}_i \sim N(0, 1)$ for $i = 16, \ldots, 40$, where $N(0, 1)$ represents the standard normal variable and $\epsilon_j \sim N(0, 0{\cdot}01)$ for $j = 1, \ldots, 15$.

Table 1. Mean entropy loss for the four simulation examples based on 1000 test iterations; bootstrap estimates of standard errors given in parentheses

Method	Simulation			
	Example 1	Example 2	Example 3	Example 4
Stepwise	0·6046 (0·139)	0·7012 (0·017)	0·0669 (0·001)	0·5088 (0·008)
Ridge	0·4215 (0·028)	0·4249 (0·023)	0·0534 (0·001)	0·2477 (0·002)
LASSO	0·4202 (0·039)	0·4941 (0·041)	0·0527 (0·001)	0·2265 (0·002)
Elastic net	0·4158 (0·033)	0·4381 (0·028)	0·0527 (0·001)	0·2280 (0·002)

In all simulations, the targets $\mathbf{y}$ were generated uniformly with probability given by (11). Example 1 and Example 2 depict scenarios where the true model is sparse and dense respectively, while Example 3 and Example 4 simulate data with grouped predictor variables.

Mean entropy loss for the four simulation scenarios is shown in Table 1. Stepwise regression had the largest mean entropy loss of all the methods tested in the four scenarios. In contrast, the three penalized regression methods performed relatively well, with the elastic net having a slight edge over ridge regression and the LASSO. LASSO performed poorly in Example 2 when compared to ridge regression and the elastic net. Ridge regression achieved the lowest entropy loss here which is not unexpected given that the predictors form a dense set. It is somewhat surprising that LASSO and ridge regression performed quite well on grouped predictor variables in Examples 3 and 4. Of the three penalized logistic regression methods, the authors recommend the elastic net as it achieves amongst the lowest entropy loss in all the scenarios tested. The elastic net is able to handle both sparse and dense predictors, as well as varying levels of correlation.

It is of interest to compare the performance of penalized regression algorithms as the number of predictors increases, while keeping the sample size constant. This roughly mimics real world data sets such as those obtained from Genome Wide Association Studies (GWAS); here the number of predictors is often much higher than the number of samples. Figure 1 depicts the mean entropy loss of elastic net, ridge regression and the LASSO as the ratio $r = (p/n)$ was increased from $r = 0, \ldots, 5$ for $(n = 50)$. For each ratio r, the regression parameters were generated with 50% sparseness; that is, approximately half of the regression parameters contained signal, while the rest were set to zero. In this example, both the elastic net and ridge regression outperformed the LASSO in terms of mean entropy loss. This is most evident when there are more predictors than samples (that is, for $r > 1$).

5.2 Real Data Examples

The performance of all methods was also examined on real data obtained from the UCI Machine Learning repository (UCI-MLR). The four data sets were: Pima Indian diabetes (250/250/268), ionosphere (100/100/151), Haberman's survival

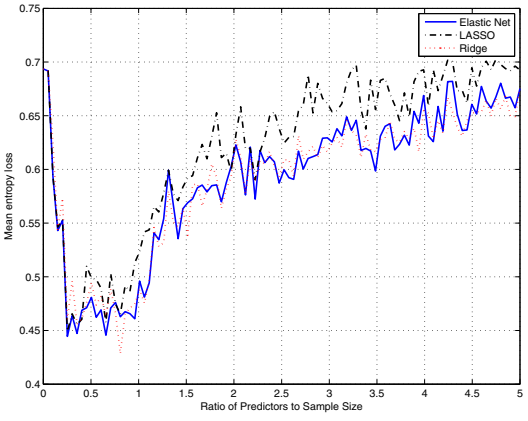
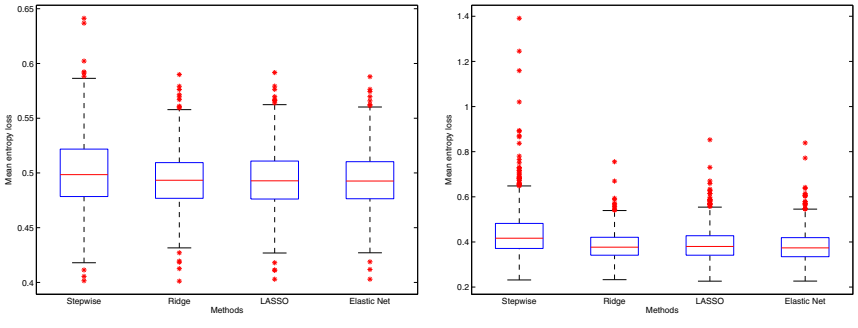
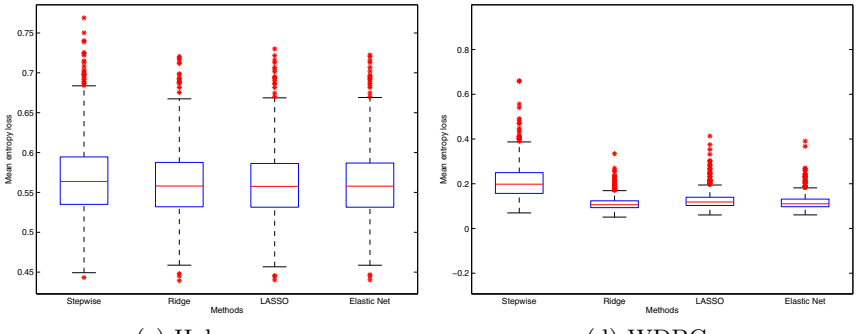


Fig. 1. Mean entropy loss for ridge, LASSO and elastic net as the ratio of parameters to sample size is increased

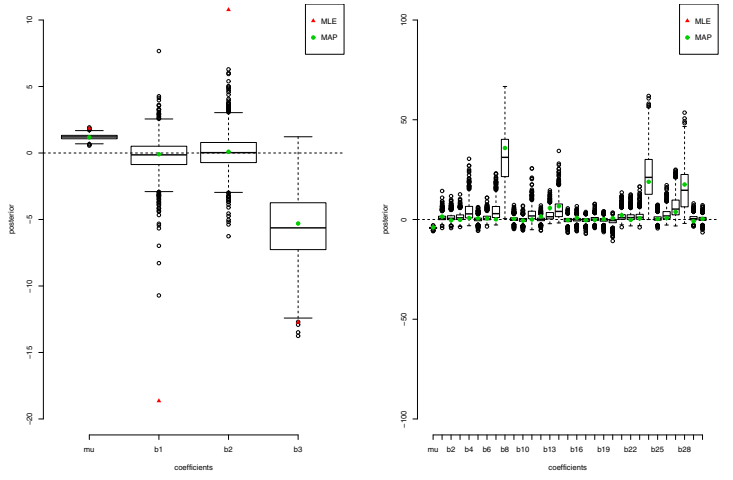


(a) Pima (b) Ionosphere



(c) Haberman (d) WDBC

Fig. 2. Mean entropy loss performance of all methods on four real data sets from the UCI Machine Learning repository



(a) Haberman (b) WDBC

Fig. 3. Regression coefficients estimated by Bayesian sampling for two real data sets

(250/250/268) and Wisconsin diagnostic breast cancer (WDBC) (100/100/369). During each iteration, a data set was randomly divided into training, validation and testing sets. The mean entropy loss for each method was recorded using only the test data; the penalty parameters were inferred with a grid search algorithm using the validation data. The entire procedure was repeated for 1000 iterations for each data set. Figure 2 depicts the mean entropy loss of the four methods tested.

As with simulated data, the penalized regression methods outperformed stepwise regression in each simulation The performance difference is most evident on the WDBC dataset which contained a moderate number of predictor variables ($p \approx 30$). In all experiments, the penalized regression methods performed roughly equally well, with ridge regression slightly outperforming LASSO and elastic net on the WDBC data.

We also briefly compared the Bayesian LASSO [4] with the standard LASSO implementation using the UCI-MLR data sets. The Bayesian LASSO outperformed the standard LASSO in terms of mean entropy loss in all four data sets; interestingly, the parameter estimates of the two methods were relatively close. Figure 3 depicts the the Bayesian coefficient estimates for Haberman's survival data and the WDBC data set. The behaviour of the LASSO shrinkage prior is clearly evident; the Bayesian estimates are much smaller than the maximum likelihood estimates and correspond to models with significantly better prediction accuracy. The maximum likelihood estimates for the WDBC data set were quite large and are not shown in Figure 3(b) for reasons of clarity. Therefore, the Bayesian LASSO should be preferred over the regular LASSO as it provides better prediction accuracy and an automatic estimate of the penalty parameter.

6 Conclusion

This paper has compared stepwise regression, ridge regression, the LASSO and the elastic net using both real and simulated data. In all scenarios, penalized logistic regression was found to be superior to stepwise regression. Of the three penalized regression methods, the elastic net is recommended as it automatically handles data with various sparsity patterns as well as correlated groups of regressors. Additionally, the Bayesian LASSO was found to be superior to the regular LASSO in terms prediction accuracy in all real data tests. This is in agreement with previous research comparing Bayesian and standard penalized regression methods on linear models.

Acknowledgments. The authors would like to thank Robert B. Gramacy for providing his Bayesian logistic regression simulation source code.

References

1. Bunea, F.: Honest variable selection in linear and logistic regression models via ℓ_1 and $\ell_1 + \ell_2$ penalization. Electronic Journal of Statistics 2, 1153–1194 (2008)
2. Cessie, S.L., Houwelingen, J.C.V.: Ridge estimators in logistic regression. Journal of the Royal Statistical Society (Series C) 41(1), 191–201 (1992)
3. Genkin, A., Lewis, D.D., Madigan, D.: Large-scale Bayesian logistic regression for text categorization. Technometrics 49(3), 291–304 (2007)
4. Gramacy, R.B., Polson, N.G.: Simulation-based regularized logistic regression (2010), arXiv:1005.3430v1
5. Hesterberg, T., Choi, N.H., Meier, L., Fraley, C.: Least angle and ℓ_1 penalized regression: A review. Statistics Survey 2, 61–93 (2008)
6. Hoerl, A., Kennard, R.: Ridge regression. In: Encyclopedia of Statistical Sciences, vol. 8, pp. 129–136. Wiley, New York (1988)
7. Holmes, C.C., Held, L.: Bayesian auxiliary variable models for binary and multinomial regression. Bayesian Analysis 1(1), 145–168 (2006)
8. James, G.M., Radchenko, P.: A generalized Dantzig selector with shrinkage tuning. Biometrika 96(2), 323–337 (2009)
9. O'Brien, S.M., Dunson, D.B.: Bayesian multivariate logistic regression. Biometrics 60(3), 739–746 (2004)
10. Scott, S.L.: Data augmentation, frequentist estimation, and the Bayesian analysis of multinomial logit models. Statistical Papers (to appear)
11. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society (Series B) 58(1), 267–288 (1996)
12. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society (Series B) 67(2), 301–320 (2005)

The Behaviour of the Akaike Information Criterion When Applied to Non-nested Sequences of Models

Daniel Francis Schmidt and Enes Makalic

The University of Melbourne
Centre for MEGA Epidemiology
Carlton VIC 3053, Australia
{dschmidt,emakalic}@unimelb.edu.au

Abstract. A typical approach to the problem of selecting between models of differing complexity is to choose the model with the minimum Akaike Information Criterion (AIC) score. This paper examines a common scenario in which there is more than one candidate model with the same number of free parameters which violates the conditions under which AIC was derived. The main result of this paper is a novel upper bound that quantifies the poor performance of the AIC criterion when applied in this setting. Crucially, the upper-bound does not depend on the sample size and will not disappear even asymptotically. Additionally, an AIC-like criterion for sparse feature selection in regression models is derived, and simulation results in the case of denoising a signal by wavelet thresholding demonstrate the new AIC approach is competitive with *SureShrink* thresholding.

1 Introduction

Every day thousands of researchers use the celebrated Akaike Information Criterion (AIC) [1] as a guide for selecting features when building models from observed data. Perhaps the most canonical example is the use of AIC to determine which features (covariates) to include in a multiple regression, which forms, for example, the basis of epidemiological and medical statistics. The AIC was derived under the assumption that the set of models under consideration (the candidate models) forms a strictly nested sequence; that is, the more complex models completely contain all of the simpler models. If we measure a model's "complexity" by the number of free parameters it possesses, a necessary (but not sufficient) requirement for this assumption to hold is that each of the candidate models possesses a unique number of free parameters.

A classic example in which this assumption is violated is subset selection of regression models; if we include all possible subsets of q features in our set of candidate models, there will be $\binom{q}{k}$ different models with exactly k free parameters. It is clear that if the number of features, q, we are considering is large then the number of models with the same number of parameters in the candidate set can be enormous.

J. Li (Ed.): AI 2010, LNAI 6464, pp. 223–232, 2010.

While the poor performance of AIC when applied to non-nested sequences of models has been noted in the literature (see for example, [2]), there appears to have been no attempts to formally quantify just how badly the AIC may perform. The primary contribution of this paper is to remedy this situation by providing a novel asymptotic upper bound quantifying the extent to which AIC may deviate from the quantity it is attempting to estimate in the setting of non-nested sequences of models. The most interesting, and worrying, finding is that the upper bound depends crucially on the maximum number of models being considered, and in the limit as the sample size $n \to \infty$ the upper bound does not converge to the usual AIC score. This implies the following critical conclusion: *that the poor performance of AIC when applied to non-nested sequences of models cannot be overcome even by obtaining large amounts of data* – the problem is tied fundamentally to the confluence of models rather than sample size. We believe this is a very important discovery with profound effects on the way the AIC should be employed in the research community.

2 Akaike's Information Criterion

The problem that the Akaike Information Criterion aims to solve is the following: we have observed n samples $\mathbf{y} = (y_1, \ldots, y_n)$ and wish to learn something about the process that generated the data. In particular, we have a set of candidate models of differing complexity which we may fit to the data. If we choose too simple a model then the predictions of future data will be affected by the bias present due to the limitations of the model; in contrast, if we choose an overly complex model then the increased variance in the parameter estimates will lead to poor predictions. The AIC aims to select the model from the candidate set that best trades off these two sources of error to give good predictions.

2.1 Models and Nested Model Sequences

It is impossible to discuss the properties of AIC and its problems when applied to non-nested sequence of models without first defining some notation. We let $\gamma \in \Gamma$ denote a statistical model, with $\boldsymbol{\theta}_\gamma \in \Theta_\gamma$ denoting a parameter vector for the model γ and Γ denoting the set of all candidate models. A statistical model γ indexes a set of parametric probability distributions over the data space; denote this by $p(\mathbf{y}|\boldsymbol{\theta}_\gamma)$. The parameter vector $\boldsymbol{\theta}_\gamma \in \Theta_\gamma$ indexes a particular distribution within the model γ. The number of free parameters possessed by a model γ (or equivalently, the dimensionality of $\boldsymbol{\theta}_\gamma$) is denoted by k_γ.

Using this notation, we can now introduce the notion of a "true" model and a "true" distribution. The true distribution is the particular distribution in the true model that generated the observed data $\mathbf{y}$. Let γ^* denote the true model, and $\boldsymbol{\theta}_*$ denote the parameter vector that indexes the true distribution. Using the shorthand notation that $p_{\boldsymbol{\theta}_\gamma}$ denotes the distribution indexed by $\boldsymbol{\theta}_\gamma$ in the model γ, we can say that $\mathbf{y} \sim p_{\boldsymbol{\theta}^*_{\gamma^*}}$.

In the context of AIC the idea of a *nested sequence* of models is very important. If a set of models form a nested sequence then they possess the special

property that a model with k free parameters can represent all of the distributions contained in all models with less than k parameters; usually, this involves setting some of the parameters to zero, though this is neither universally the case, nor a requirement. The following are two important properties possessed by nested sequences of models.

Property 1. Each of the models in a nested sequence of models has a unique number of free parameters.

Property 2. If the true model γ^* is part of a nested sequence of models, then for all models γ with $k_\gamma > k_{\gamma^*}$ (i.e., with more free parameters) there is a parameter vector $\boldsymbol{\theta}_\gamma \in \Theta_\gamma$ that indexes the same distribution as the "true" distribution $p_{\boldsymbol{\theta}_*}$.

Let this parameter vector be denoted by the symbol $\boldsymbol{\theta}^*_\gamma$.

In words, this says that if the true distribution can be represented by the model in the nested sequence with k parameters, then it can also be exactly represented by all the models with more than k parameters. Thus, the "true" model is simply the model with the least number of parameters that can represent the true distribution. An example will illustrate the concepts presented in this section.

Example: Polynomial Models. Consider the class of normal regression models, where the mean is specified by a polynomial of degree k. If the maximum degree is q, the model class index $\gamma \in \{0, 1, \ldots, q\}$ denotes the degree of the polynomial; i.e., $\gamma = k$ specifies a polynomial of the form

$$y = a_0 + a_1 x + a_2 x^2 + \ldots a_k x^k + \varepsilon$$

with ε normally distributed with variance τ. The polynomial model indexed by γ has $k_\gamma = \gamma + 2$ free parameters (including the noise variance) given by $\boldsymbol{\theta}_\gamma = (a_0, \ldots, a_\gamma, \tau)$, with the parameter space $\Theta_\gamma = \mathbb{R}^{k+1} \times \mathbb{R}_+$. The models form a *nested sequence* as a polynomial of degree k can represent any polynomial of degree $j < k$ by setting $a_{j+1}, \ldots, a_k = 0$; for example, a quintic polynomial can represent a cubic polynomial by setting $a_4 = a_5 = 0$.

2.2 Model Fitting and Goodness of Fit

There are many ways of fitting a model γ to the observed data (often called "point estimation"); a powerful and general procedure is called *maximum likelihood* (ML), and it is this process that is integral to the derivation of the AIC. Maximum likelihood fitting simply advocates choosing the parameter vector $\boldsymbol{\theta}_\gamma$ for a chosen model γ such that the probability of observed data $\mathbf{y}$ is maximised

$$\hat{\boldsymbol{\theta}}_\gamma = \underset{\boldsymbol{\theta}_\gamma \in \Theta_\gamma}{\arg\max} \{p(\mathbf{y}|\boldsymbol{\theta}_\gamma)\} \tag{1}$$

For a model selection criterion to be useful it must aim to select a model from the candidate set that is close, in some sense, to the truth. In order to measure

how close the fitted approximating model $\hat{\boldsymbol{\theta}}_\gamma$ is to the generating distribution $\boldsymbol{\theta}_*$, one requires a distance measure between probability densities. A commonly used measure of distance between two models, say $\boldsymbol{\theta}_*$ and $\hat{\boldsymbol{\theta}}_\gamma$ is the directed Kullback–Leibler (K–L) divergence [3], given by

$$\Delta(\boldsymbol{\theta}_* || \hat{\boldsymbol{\theta}}_\gamma) = \mathrm{E}_{\boldsymbol{\theta}_*} \left[\log \frac{p(\mathbf{y}|\boldsymbol{\theta}_*)}{p(\mathbf{y}|\hat{\boldsymbol{\theta}}_\gamma)} \right] \tag{2}$$

where the expectation is taken with respect to $\mathbf{y} \sim p_{\boldsymbol{\theta}_*}$. The directed K–L divergence is non-symmetric and strictly positive for all $\hat{\boldsymbol{\theta}}_\gamma \neq \boldsymbol{\theta}_*$. Defining the function

$$d(\boldsymbol{\theta}_*, \hat{\boldsymbol{\theta}}_\gamma) = 2\mathrm{E}_{\boldsymbol{\theta}_*} \left[\log 1/p(\mathbf{y}|\hat{\boldsymbol{\theta}}_\gamma) \right] \tag{3}$$

the K–L divergence may be written as

$$2\Delta(\boldsymbol{\theta}_* || \hat{\boldsymbol{\theta}}_\gamma) = d(\boldsymbol{\theta}_*, \hat{\boldsymbol{\theta}}_\gamma) - d(\boldsymbol{\theta}_*, \boldsymbol{\theta}_*) \tag{4}$$

The first term on the right hand side of (4) is generally known as the cross-entropy between $\boldsymbol{\theta}_*$ and $\hat{\boldsymbol{\theta}}_\gamma$, while the second is known as the entropy of $\boldsymbol{\theta}_*$. The use of the Kullback–Leibler divergence can be justified by both its invariance to the parameterisation of the models (as opposed to Euclidean distance, for example) as well as its connections to information theory.

2.3 Akaike's Information Criterion

Ideally, one would rank the candidate models in ascending order based on their K–L divergence from the truth, and select the model with the smallest K–L divergence as optimal. However, this procedure requires knowledge of the true model and is thus not feasible in practice. Even though the truth is not known, *one may attempt to construct an estimate of the K–L divergence based solely on the observed data*. This idea was first explored by Akaike in his groundbreaking paper [1] in the particular case of a nested sequence of candidate models. Akaike noted that the negative log-likelihood serves as a downwardly biased estimate of the average cross entropy (the cross-entropy risk), and subsequently derived an asymptotic bias correction. The resulting Akaike Information Criterion (AIC) advocates choosing a model, from a nested sequence of models, that minimises

$$\mathrm{AIC}(\gamma) = 2 \log 1/p(\mathbf{y}|\hat{\boldsymbol{\theta}}_\gamma) + 2k_\gamma \tag{5}$$

where $\hat{\boldsymbol{\theta}}_\gamma$ is the maximum likelihood estimator for the model γ and the second term is the bias correction. Under suitable regularity conditions [4], and assuming that the fitted model γ is at least as complex as the truth (i.e., the true distribution is contained in the distributions indexed by the model γ), the AIC statistic can be shown to satisfy

$$\mathrm{E}_{\boldsymbol{\theta}^*} \left[\mathrm{AIC}(\gamma) \right] = \mathrm{E}_{\boldsymbol{\theta}^*} \left[d(\boldsymbol{\theta}^*, \hat{\boldsymbol{\theta}}_\gamma) \right] + o_n(1) \tag{6}$$

where $o_n(1)$ denotes a term that vanishes as the sample size $n \to \infty$. In words, (6) states that the AIC statistic is, up to a constant, an unbiased estimator of twice the Kullback–Leibler risk (average Kullback–Leibler divergence from the truth) attained by a particular model γ; that is, for sufficiently large sample sizes, the AIC score is *on average* equal to the average cross–entropy between the truth and the maximum likelihood estimate for the fitted model γ. Although the AIC estimates the cross–entropy risk rather than the complete Kullback–Leibler risk, the omitted entropy term $d(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*)$ does not depend on the fitted model γ and will thus have no effect on the ranking of models by their AIC scores. The selection of a candidate model using AIC is therefore equivalent to choosing one with the lowest estimated Kullback–Leibler risk.

In the case of non-nested model sequences, the number of candidate models with k parameters may be greater than one and the downward bias of the negative log-likelihood is greater than the AIC model structure penalty. Problematically, this extra source of additional bias remains even as the sample size $n \to \infty$. The next section derives a novel upper-bound on this additional bias under certain conditions.

3 The Bias in AIC for Multiple Selection

The main result of this paper is an expression for the additional downward bias that is introduced when $q_k > 1$. Let

$$\Gamma_k = \{\gamma \in \Gamma : k_\gamma = k\}$$

denote the set of all candidate models with k parameters, with $q_k = |\Gamma_k|$ being the number of candidate models with k parameters. In the case of a nested sequence of models, $q_k = 1$ for all k. Then, let

$$\hat{m}_k = \underset{m \in \Gamma_k}{\arg\min} \left\{ \log 1/p(\mathbf{y}|\hat{\boldsymbol{\theta}}_m) \right\} \tag{7}$$

denote the candidate model with k parameters with the smallest negative log-likelihood. We can now recast the model selection problem as one of selecting between the *best* of the k parameter models, i.e. we limit our candidates to the new set of L fitted models

$$\Gamma' = \left\{ \hat{\boldsymbol{\theta}}_{\hat{m}_1}, \ldots, \hat{\boldsymbol{\theta}}_{\hat{m}_L} \right\} \tag{8}$$

Assuming the following holds

1. The true model γ^* has no free parameters
2. All candidate models $\gamma \in \Gamma$ contain the true distribution $p_{\boldsymbol{\theta}_*}$ as a particular element, i.e., *all candidate models are overfitting*. Let the parameter vector that indexes the true distribution for the model γ be denoted by $\boldsymbol{\theta}^*_\gamma$
3. The maximum likelihood estimator converges to the truth, $\hat{\boldsymbol{\theta}}_\gamma \to \boldsymbol{\theta}^*_\gamma$ as $n \to \infty$, and is asymptotically normally distributed, $\hat{\boldsymbol{\theta}}_\gamma \sim N(\boldsymbol{\theta}^*_\gamma, \mathbf{J}^{-1}(\boldsymbol{\theta}^*_\gamma))$

4. All candidate models of k parameters are independent; that is,

$$\log \frac{p(\mathbf{y}|\boldsymbol{\theta}_*)}{p(\mathbf{y}|\hat{\boldsymbol{\theta}}_m)}, \quad m \in \Gamma_k$$

are independent random variates.

Theorem 1: Under the above conditions we have

$$2\mathrm{E}_{\boldsymbol{\theta}^*}\left[\log 1/p(\mathbf{y}|\hat{\boldsymbol{\theta}}_{\hat{m}_k})\right] + 2\alpha(k, q_k) = \mathrm{E}_{\boldsymbol{\theta}_*}\left[d(\boldsymbol{\theta}_*, \hat{\boldsymbol{\theta}}_{\hat{m}_k})\right] + o_n(1) \tag{9}$$

where

$$\alpha(k, q_k) = \mathrm{E}_{\chi^2_k}\left[\max\{z_1, \ldots, z_{q_k}\}\right] \tag{10}$$

and $z_1, \ldots, z_{q_k}$ are independently and identically distributed χ^2_k variates with k degrees of freedom.

Proof: Following the procedure in [5] the cross-entropy risk can be written

$$\begin{aligned}\mathrm{E}_{\boldsymbol{\theta}_*}\left[d(\boldsymbol{\theta}_*, \hat{\boldsymbol{\theta}}_m)\right] &= \mathrm{E}_{\boldsymbol{\theta}_*}\left[d(\boldsymbol{\theta}_*, \hat{\boldsymbol{\theta}}_m)\right] - d(\boldsymbol{\theta}_*, \boldsymbol{\theta}_*) \\ &\quad + d(\boldsymbol{\theta}_*, \boldsymbol{\theta}_*) - 2\mathrm{E}_{\boldsymbol{\theta}_*}\left[\log 1/p(\mathbf{y}|\hat{\boldsymbol{\theta}}_m)\right] \\ &\quad + 2\mathrm{E}_{\boldsymbol{\theta}_*}\left[\log 1/p(\mathbf{y}|\hat{\boldsymbol{\theta}}_m)\right]\end{aligned} \tag{11}$$

From regularity conditions the following approximations hold

$$2\log 1/p(\mathbf{y}|\boldsymbol{\theta}_*) + 2\log p(\mathbf{y}|\hat{\boldsymbol{\theta}}_m) = (\boldsymbol{\theta}^*_m - \hat{\boldsymbol{\theta}}_m)'\mathbf{H}(\hat{\boldsymbol{\theta}}_m, \mathbf{y})(\boldsymbol{\theta}^*_m - \hat{\boldsymbol{\theta}}_m) + o(k) \tag{12}$$

$$d(\boldsymbol{\theta}_*, \hat{\boldsymbol{\theta}}_m) - d(\boldsymbol{\theta}_*, \boldsymbol{\theta}_*) = (\boldsymbol{\theta}^*_m - \hat{\boldsymbol{\theta}}_m)'\mathbf{J}(\boldsymbol{\theta}^*_m)(\boldsymbol{\theta}^*_m - \hat{\boldsymbol{\theta}}_m) + o(k) \tag{13}$$

where

$$\mathbf{H}(\hat{\boldsymbol{\theta}}_m, \mathbf{y}) = \left[\left.\frac{\partial^2 \log 1/p(\mathbf{y}|\boldsymbol{\theta}_m)}{\partial\boldsymbol{\theta}_m \partial\boldsymbol{\theta}'_m}\right|_{\boldsymbol{\theta}_m = \hat{\boldsymbol{\theta}}_m}\right], \quad \mathbf{J}(\boldsymbol{\theta}^*) = \left[\left.\frac{\partial^2 \Delta(\boldsymbol{\theta}^*, \boldsymbol{\theta})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\right|_{\boldsymbol{\theta} = \boldsymbol{\theta}^*}\right]$$

are the observed and expected Fisher information matrices respectively. Denote the right hand side of (12) and (13) by a_m and b_m respectively. The first term, a_m, is twice the decrease in the negative log-likelihood due to fitting a model $\hat{\boldsymbol{\theta}}_m$, and the second term, b_m, is twice the K–L divergence between the generating model $\boldsymbol{\theta}^*$ and the fitted model $\hat{\boldsymbol{\theta}}_m$. Since there are q_k models with k parameters, there are q_k random variables a_m and b_m.

Selecting the model with k parameters that minimises the negative log-likelihood is equivalent to solving

$$\hat{m}_k = \arg\max_{m \in \Gamma_k} \{a_m\}$$

Then we have

$$2\mathrm{E}_{\boldsymbol{\theta}^*}\left[\log 1/p(\mathbf{y}|\boldsymbol{\theta}^*) - \log 1/p(\mathbf{y}|\hat{\boldsymbol{\theta}}_{\hat{m}_k})\right] = \mathrm{E}_{\boldsymbol{\theta}^*}\left[a_{\hat{m}_k}\right] + o_n(1) \tag{14}$$

$$\mathrm{E}_{\boldsymbol{\theta}^*}\left[d(\boldsymbol{\theta}^*, \hat{\boldsymbol{\theta}}_{\hat{m}_k}) - d(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*)\right] = \mathrm{E}_{\boldsymbol{\theta}^*}\left[b_{\hat{m}_k}\right] + o_n(1) \tag{15}$$

For large n, the random variables satisfy $a_m = b_m + o_n(1)$ and therefore coincide. From the properties of the maximum likelihood estimator $\mathbf{H}(\hat{\boldsymbol{\theta}}_m, \mathbf{y}) \to \mathbf{J}(\boldsymbol{\theta}^*_m)$ as $n \to \infty$, rendering the quadratic forms in (12) and (13) identical. Furthermore, a_m converge to centrally distributed χ^2_k variates with k degrees of freedom. Thus,

$$\mathrm{E}_{\boldsymbol{\theta}^*}\left[a_{\hat{m}_k}\right] = \mathrm{E}\left[\max\{z_1, \ldots, z_{q_k}\}\right] \tag{16}$$

where $z_1, \ldots, z_{q_k}$ are independently and identically distributed χ^2_k variates with k degrees of freedom, with an identical expression for $\mathrm{E}_{\boldsymbol{\theta}^*}\left[b_{\hat{m}_k}\right]$. Substituting these expectations into the expression for $\mathrm{E}_{\boldsymbol{\theta}^*}[d(\boldsymbol{\theta}^*, \hat{\boldsymbol{\theta}}_{\hat{m}_k})]$ given by (11) completes the proof. □

4 Discussion and Impact

We now discuss the impact of Theorem 1. In words, the result states that if we consider more than one candidate model with the same number of parameters, say k, then the usual AIC complexity penalty of $2k$ (or alternatively, the bias correction) for these models will be insufficient. A further negative result is that under the above conditions, the required bias correction depends on the number of models with k parameters, q_k, and k, but not on the sample size n, and will not disappear even as $n \to \infty$. The primary effect an underestimation of bias will have in practice is to lead to an increased probability of overfitting.

As an example, consider the situation in which the "true" model, γ^*, has no free parameters, and we are considering as alternatives, based on regular AIC scores, a set of $q_1 \geq 1$ "independent" models with one free parameter. In the usual case of a nested sequence of models $q_1 = 1$, and noting that twice the difference in log-likelihoods between the fit of γ^* and the alternative one parameter model is approximately χ^2_1 distributed, we determine that AIC has approximately a 16% probability of eroneously preferring the one parameter model (overfitting). This probability will increase with increasing q_k: using the results of Theorem 1, we see that if $q_k > 1$ then twice the difference in negative log-likelihoods between the initial model we fit, γ_*, and the best of the one parameter models, $\gamma_{\hat{m}_1}$, is distributed as per the maximum of q_1 χ^2_1 variates with one degree of freedom. Using standard results on distributions of order statistics [6], we can compute the probability of overfitting in this scenario for various values of q_1; these are summarised in Table 1. It is clear that even if we consider only four models with $k = 1$ parameters, the probability of overfitting is almost one half, and that it rapidly rises towards one as q_1 increases. This demonstrates just how poorly regular AIC may perform when applied to a non-nested sequence of models.

Table 1. Probability of AIC overfitting by one parameter for various values of q_1

q_1	1	2	3	4	5	8	10	15	25	50	100
P(overfit)	0·157	0·290	0·402	0·496	0·575	0·746	0·819	0·923	0·986	0·999	1·000

4.1 Theorem 1 as an Upper Bound

The most restrictive assumption used by Theorem 1 is the requirement that the models be "independent" (Assumption 4 in Section 3). For many models, this will not be the case; a simple example is "all subsets" regression models, where many of the subsets of two or more features will have several features in common. If one feature is strongly associated with the target, then all subsets containing this feature will reduce the negative log-likelihood by a similarly large amount, i.e., the a_m variates from Theorem 1 will be correlated . However, even in the case that Assumption 4 is violated, the result of Theorem 1 offers a novel *upper bound*: noting that if $\{w_1, \ldots, w_q\}$ are q correlated variates and $\{z_1, \ldots, z_q\}$ are uncorrelated variates, with both sets of variates possessing the same marginal distribution, then

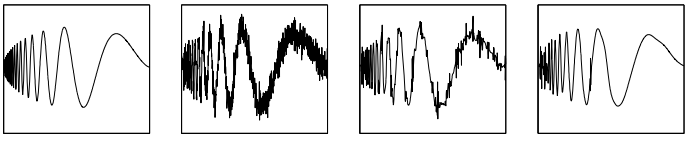
$$\mathrm{E}\left[\max\left\{w_1, \ldots, w_q\right\}\right] < \mathrm{E}\left[\max\left\{z_1, \ldots, z_q\right\}\right]$$

so that the asymptotic bias correction term in this case will be less than $2\alpha(k, q)$. Thus, the result in Theorem 1 acts as an upper bound on the asymptotic bias correction.

5 Forward Selection of Regression Features

A common application of model selection procedures in machine learning and data mining is *feature selection*. Here, one is presented with many features (explanatory variables, covariates) and a single target variable $\mathbf{y}$ we wish to explain with the aid of some of these features. The AIC criterion is often used to determine if a feature is useful in explaining the target; this is a type of "all subsets" regression, in which any combination of features is considered plausible *a priori*, the data itself being used to determine whether the features are significant or statistically useful. Unfortunately, as the number of features may often be very large, the results of Section 3 suggest that the usual AIC is inappropriate, and choosing features by minimising an AIC score will generally lead to large numbers of "spurious" features being included in the final model. We propose a forward-selection AIC-like procedure, called AIC_m, based on the results of Theorem 1. Forward selection of features acts by iteratively enlarging the current model to include the feature that most improves the fit, and produces a type of nested sequence of models; unfortunately, the sequence is determined by the available data rather than *a priori* and so violates the usual AIC conditions.

The main idea behind our procedure is to note that, with high probability, the important non-spurious features will yield the best improvements in fit and be included before the spurious features. Thus, if there are k^* non-spurious features, the first k^* subsets created by the forward selection procedure will, with high probability, be the same irrespective of the random noise corrupting



(a) "Doppler" (b) Noisy (c) SURE (d) AIC_m

Fig. 1. Denoising of the "Doppler" Test Signal by *SureShrink* and AIC_m

our data, and thus form a usual nested sequence of models. However, once all k^* non-spurious features have been included, *the remaining* $(q - k^*)$ *subsets depend entirely on the random noise and form a non-nested sequence of models*; the results of Theorem 1 may be used to avoid selecting these spurious features.

The AIC_m procedure may be summarised as follows. Let $\gamma[k]$ denote the set of the q features included at step k, so that $\gamma[0] = \emptyset$, i.e., we start with an empty model and let $\bar{\gamma}[k] = \{1, \ldots, q\} - \gamma[k]$ denote the set of features not in $\gamma[k]$. Then, for $k = 0$

1. Find the unused feature that most decreases the negative log-likelihood
$$\gamma[k+1] = \underset{j \in \bar{\gamma}[k]}{\arg\min} \left\{ \log 1/p(\mathbf{y}|\hat{\boldsymbol{\theta}}_{\gamma[k] \cup j}) \right\}$$
2. If $\left(\log 1/p\left(\mathbf{y}|\hat{\boldsymbol{\theta}}_{\gamma[k]}\right) - \log 1/p\left(\mathbf{y}|\hat{\boldsymbol{\theta}}_{\gamma[k+1]}\right) \right) < (\alpha(1, q-k)+1)/2$ the feature is rejected and algorithm terminates
3. $k \leftarrow k + 1$; if $k = q$, algorithm terminates, otherwise go to Step 1.

The threshold for rejection is based on two observations; the first is that even if all $(q - k)$ remaining features at step k were spurious we would still expect to see, on average, an improvement in negative log-likelihood of $\alpha(1, q - k)/2$ from the best one amongst them (from the expectation of the a_m variates in Theorem 1). This accounts for the first term in the threshold. The second term arises by noting that if the improvement exceeds the first threshold, we are deciding the feature is non-spurious; at this point, we can use the regular AIC penalty of $1/2$ a unit to account for the variance introduced by estimating the extra parameter.

5.1 Application: Signal Denoising by Wavelet Thresholding

An interesting example of regression in which the number of features is very large is denoising or smoothing of a signal using orthonormal basis functions called "wavelets". An excellent discussion of wavelets, and their properties for smoothing, can be found in [7]), and it is from this paper we take our four test signals. These signals, called "Bumps", "Blocks", "HeaviSine" and "Doppler" are benchmarks in the wavelet literature and are designed to caricature various types of signals found in real applications.

We tested our AIC_m procedure on the wavelet smoothing problem by first applying the discrete wavelet transform to the noise corrupted versions of the test

Table 2. Squared prediction errors for denoising of the Donoho-Johnston test signals

	"Bumps"		"Blocks"		"HeaviSine"		"Doppler"	
SNR	Sure	AIC_m	Sure	AIC_m	Sure	AIC_m	Sure	AIC_m
1	0·4129	0·3739	0·3140	0·2302	0·2164	0·0315	0·2686	0·0963
10	0·0578	0·0478	0·0530	0·0498	0·0239	0·0079	0·0355	0·0157
100	0·0070	0·0055	0·0066	0·0055	0·0033	0·0015	0·0047	0·0020

signals, and then using our criterion to determine which wavelets (our "features") to include, the maximum number of wavelets possible being restricted to $n/2$ to ensure that the asymptotic conditions are not violated. The closeness of the resulting smoothed signal to the true signal was assessed using average mean squared error, and our AIC_m procedure was compared against the well known *SureShrink* algorithm [7]. Three levels of signal-to-noise ratio (SNR) (the ratio of signal variance to noise variance) were used, and for each combination of test signal and SNR level, the two criterion were tested one thousand times. The mean squared errors presented in Table 2 clearly demonstrate the effectiveness of the AIC_m procedure; in contrast, applying regular AIC resulted in the maximum number of $n/2$ wavelets being included in every case, with correspondingly poor performance. Figure 1 demonstrates the difference in performance between AIC_m and *SureShrink* for the "Doppler" signals at an SNR of ten; the AIC_m smoothing is visually superior to that obtained by *SureShrink*.

6 Conclusion

This paper examined the failings of AIC as a model selection criterion when the set of candidate models forms a non-nested sequence. The main contribution was a novel theorem quantifying the bias in the regular AIC estimate of the Kullback–Leibler risk, which demonstrated that this bias may not be overcome even as the sample size $n \to \infty$. This result was used to derive an AIC-like procedure for forward selection in regression models, and simulations suggested the procedure was competitive when applied to wavelet denoising.

References

1. Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control 19(6), 716–723 (1974)
2. Hurvich, C.M., Tsai, C.L.: A crossvalidatory AIC for hard wavelet thresholding in spatially adaptive function estimation. Biometrika 85, 701–710 (1998)
3. Kullback, S., Leibler, R.A.: On information and sufficiency. The Annals of Mathematical Statistics 22(1), 79–86 (1951)
4. Linhart, H., Zucchini, W.: Model Selection. Wiley, New York (1986)
5. Cavanaugh, J.E.: A large-sample model selection criterion based on Kullback's symmetric divergence. Statistics & Probability Letters 42(4), 333–343 (1999)
6. Cramér, H.: Mathematical methods of statistics. Princeton University Press, Princeton (1957)
7. Donoho, D.L., Johnstone, I.M.: Adapting to unknown smoothness via wavelet shrinkage. Journal of the Amer. Stat. Ass. 90(432), 1200–1224 (1995)

Unbounded Knowledge Acquisition Based upon Mutual Information in Dependent Questions

Tony C. Smith and Chris van de Molen

Department of Computer Science, University of Waikato, Hamilton, New Zealand
tcs@cs.waikato.ac.nz

Abstract. This paper describes an experimental system for knowledge acquisition based on a general framework exemplified in the game of twenty questions. A sequence of propositional questions is put to the user in an attempt to uncover some hidden concept, and the answers are used to expand and refine the system's knowledge of the world. Previous systems adopting this framework typically represent knowledge as a matrix of truth values or weights that relate entities to attributes—such that if the hidden concept is "a bird", for example, then the answer to a question about whether the target entity can fly is based on the extent to which "flying" is generally attributable to "a bird" as measured by the value in the matrix element indexed by the attribute-entity pair. Our system adopts a subtly different approach wherein knowledge is a measure of the extent to which answers to pairs of questions are co-dependent. Thus, knowledge about birds being able to fly is captured by the mutual information in the answers to a pair of questions like "Can it fly?" and "Is it a bird?". We present a case that this offers a practical and epistemologically sound basis for acquiring knowledge.

Keywords: knowledge acquisition, mutual information, twenty questions.

1 Introduction

There is a long tradition in computer science whereby the so-called *intelligence* of a system is in some way estimated from its ability to answer questions. The Turing test is the classic example, but widespread active research into automatic answer extraction and open-domain question answering for information retrieval tasks indicates a continuing interest in creating programmes that can respond to inquiries[1,4,3]. In theories of epistemology, however, the ability to *generate* questions is also important to intelligence. As any four-year old child will demonstrate, asking questions is one of the most powerful ways to acquire new knowledge.

This paper describes an investigation into the development of a computer system that can refine and expand its factual knowledge of the world (without limit) by asking carefully selected propositional questions, and processing the answers received. The underlying methodology follows after the Interrogative Model of

J. Li (Ed.): AI 2010, LNAI 6464, pp. 233–242, 2010.

Inquiry (I-Model) developed by Jaako Hintikka [5,6], wherein new knowledge and understanding emerge through a dynamic process of inquiry comprised from two levels of questions: a principal cognitive goal (or "big question") whose answer is obtained by asking a series of subordinate questions. The approach is perhaps best understood with the example of the game of twenty-questions, where up to nineteen qualifying questions are posed in order to set up (with as high a probability of getting an affirmative reponse as is possible) the twentieth question, which ultimately satisfies the overall objective of the inquiry.

The system described in this paper more or less emulates the game of twenty-questions, but without any limit imposed as to the number of questions that may be asked. The user is expected to think of something, and the computer adaptively tries to guess what it is by selecting an effective sequence of yes/no questions. Each answer is used to help the system select the next best question so as to minimise the number of questions needed to satisfy the overall goal of the inquiry. After each episode (i.e. after the "answer" becomes known at the end of the inquiry) answers are subsequently used to update the system's knowledge base so that future iterations of the game are played more effectively.

Many implementations already exist for playing twenty questions (a particularly good one being `http://www.20q.net`) and, in that respect, merely creating another would be largely uninteresting. Indeed, at the outset, our motivation had more to do with finding a way to expand a system's knowledge base whenever the target concept being guessed turned out to be unknown to the system—in this way, using twenty questions as a basis for knowledge acquistion. The initial idea was that if the computer ultimately could not guess the hidden concept, the user would be asked to name it then the computer would add it to its knowledge base, along with the information acquired as to the entity's attributes as reflected by the questions and answers that had been obtained during the interrogation.

Many interesting epistemological issues arose during the development of this *knowledge acquisition* system—specifically, matters that arose while trying to 1) determine how knowledge could be represented, updated, and expanded effectively through the process of inquiry, and 2) decide how subordinate questions should be selected to achieve the desired goal most efficiently. We report here how we ultimately overcame these design problems in an unusual way: first, by representing tacit knowledge directly as questions, and second, by representing implicit knowledge as the mutual information expressed in the answers to all pairs of questions.

2 The Knowledge Base

Twenty-questions (in the unlikely event that you don't know this already) involves uncovering the "hidden thing" by asking up to twenty questions that have a *yes* or *no* answer (leaving aside for the moment the problem of multivalued-truth and corresponding answers such as *sometimes*, *seldom* or *usually*). For example, the inquirer may ask "Is it a person?" first, and then following an affirmative response might ask "Is this person still alive?", and so on. Obviously

twenty such questions allow the inquirer to uniquely identify just over a million (2^{20}) distinct things. Randomly chosen questions generally don't result in an effective strategy for isolating the objective fact, so a good player chooses questions which divide the search space as nearly in half as is possible (taking into account possible perceived biases the answerer might be expected to have in selecting their hidden thing).

To design the knowledge base for a computer system that will play the part of the inquirer in this guessing game, we need a data structure that will let the system identify which unasked questions best partitions the search space and thus represents the best candidate to be asked next. For an adaptive acquisition system, the data structure must be able to record questions and answers so that its knowledge can be better conditioned for more effective use in the future, and it must allow new questions to be added to expand its knowledge base. In this section we show how an entity-attribute truth table (perhaps the most obvious choice of data structure) is unsuitable for an adaptive system, and describe a question-question correspondence matrix that works well as an alternative. We further discuss how the question-question matrix better reflects epistemological theory.

2.1 The Entity-Attribute Truth Table

One obvious way to represent propositional knowledge for a question-asking system is with a simple two-dimensional matrix such as that pictured in Figure 1, where each row corresponds to an attribute and each column to an entity, and each cell is set to either *true* or *false* (or to some *real number* for multivalued-truth representation) to reflect whether or not (or to what degree) the entity exhibits that attribute. To select the first question to ask, the system analyses each attribute and chooses the one that best partitions the search space—that is, the one whose true/false count ratio for all entities is closest to one. (In the case of a draw, one of the candidates can be chosen at random.) The attribute is formulated into a question and posed to the user. Once an answer is obtained, all attributes of entities whose truth value does not match the response to the question are eliminated from the set of candidates for the next question, and the process is repeated over the eligible portion of the table. The process stops when either 1) a single entity is isolated, in which case it is guessed as the *hidden fact*, or 2) no more attributes are left untested that can differentiate the remaining candidate entities, in which case each entity must be guessed in turn, or 3) all entities are eliminated as possible targets, in which case the system does not know the answer and must add a new column to the table and somehow label it with the new entity.

There are several observations to be made about this structure which indicate a flaw as to its completeness, and which suggest how it can be improved. First, note that if the first condition arises (i.e. a single entity is isolated by the response to a question), as when "is it alive?" is asked based on the knowledge matrix in Figure 1 and is rewarded with a negative reply, then the system must break out of the process of selecting an attribute to query (i.e. a row of the matrix) and

	fruitbat	eagle	tiger	rock	
is alive	T	T	T	F	
flies	T	T	F	F	
lays eggs	F	T	F	F	

Fig. 1. An entity-attribute truth table knowledge base

instead generate a final question testing the principal cognitive goal—turning the entity label for the column into a question. That is, guessing that the hidden thing is a particular entity effects the same outcome as if one had instead had access to an attribute-testing question that asked if the hidden thing has the attribute of *being that particular entity.* The identity of the entity can be viewed as just one more attribute to be tested. Given the attribute/entity format of the table, this *entity-ness* attribute value would be set to true for the target entity and false for all others. Leaving aside for the moment the question of what might be gained from the addition of an entity-testing question as a row in the matrix, we note that once the answer is obtained then it follows the third termination condition arises (i.e. all entities are eliminated as possible targets).

We can similarly reduce the second termination condition to an instance of the third by creating entity-testing questions for each remaining candidate entity. That is, the list of possible entities for which *no further qualifying attributes remain untested* could each be translated into entity-testing questions of the form "does the thing you are thinking of have the attribute of being Entity A?", where each remaining entity that is not Entity A has some implicit *Entity A-ness* attribute set to false. The moment an affirmative response is obtained, all entities are eliminated as possible candidates because they would not have the entity-ness attribute marked as existant, by definition. Hence, asking a question that tests the principal cognitive goal can be made identical to asking a subordinate question.

But if a column label (i.e. entity name) is converted into a row (i.e. an entity-ness attribute question), then we must formulate another meaning for the truth-value of a cell, since it is no longer indexed in the vertical dimension by the entity. What sort of data structure would allow us to treat *specific entity-ness* uniformly with all other attributes? Before we propose one, it is worthwhile noting some other shortcomings of the entity-attribute truth table. Note, for example, that in the case of the second stopping condition (with multiple candidate entities), it is possible for the system to guess the target entity with its first direct question. If entity-ness is just another attribute, how can the system detect it has just reached its principal cognitive goal? How does it know it hasn't merely gained some additional qualifying information, and that it shouldn't now go on and ask other entity-ness questions? What justification is needed to prevent it from

continuing to ask questions; testing all potential entity-nesses as it pursues isolation of a target entity? Indeed, the user's intended target might be guessed at any time (perhaps with the first question), regardless of whether additional attributes remain that could be tested. How does it know to stop?

One might suggest that the system only be allowed to succeed when the first stopping condition is reached with an affirmative reply; all else is treated as failure. But this is unsatisfactory in some circumstances. For example, imagine that the last testable attribute is whether or not the user is thinking of a tiger, and the user believes that the final target concept is actually more specifically a Bengal tiger; then the system has failed in some sense. It would pose *tiger* as a final guess, and after the user provides the answer, the system would have to assume it has reached the cognitive goal or hit the third stopping condition. Ignoring this problem for now, imagine the system somehow realises it has reached the latter state and engages its knowledge acquisition procedure (which we shall present shortly), and thereby manages to add a question about *Bengal-tiger-ness* so that it will be able to guess correctly the next time this entity comes up as the target. This only puts-off the problem, for in a subsequent iteration of the game the user may have as the hidden target a female Bengal tiger? Or a female Bengal tiger in the Sydney zoo? Or a female tiger in the Sydney zoo that just had cubs? And so on. The more general problem is: what (apart from the third stopping condition after a negative reply) allows a questioner to know when it has asked the last question necessary to end the inquiry?

Of course this problem also exists for humans when they play twenty-questions among themselves. If the target is uncovered in fewer than twenty questions, some signal is given by the answerer so that the guesser does not continue their quest. We simply stipulate the same courtesy be given to the computer system. The user should respond with something other than just "yes" when the final target is guessed—perhaps by saying "you guessed it" instead.

2.2 Acquisition

Another difficulty for the entity-attribute truth table arises when it comes time to expand its knowledge base. Actually, there are two problems. When the system stops due to condition two, it would be useful if the system could add another attribute row to its table—an attribute that will in future allow it to differentiate between competing candidate entities. When the system stops under condition three, a new column must be added and labeled with the previously unknown entity. In the absence of quite sophisticated natural language processing software, the only way to achieve this is to manually alter the computer programme (specifically, its database of knowledge). Of course we may already have to assume such sophisticated NLP software in order to take attribute labels and entity names and generate well-formed questions about them.

An entity-attribute truth table was initially used for the knowledge base in this system. But trying to solve all of the problems outlined above proved a considerable challenge, and part way through the study it was realised that a different kind of data structure would obviate all of these shortcomings and make

it a lot easier to concentrate on the fundamental goal of using the interrogative model of inquiry as the basis for knowledge acquisition. The key insight was to represent knowledge not as relationships between attibutes and entities at all, but simply as relationships between questions.

2.3 A Question-Question Correspondence Matrix

One easy way to eliminate the natural language processing problems entailed by having to generate questions from attribute labels and entity names, and to expand the knowledge base with new attributes and entities without recoding, is simply to represent all tacit knowledge explicitly as wellformed questions.

We noted in the previous section that one could view the act of asking whether some particular entity is the "final target" as the act of asking whether the target has the attribute of some particular entity-ness. To do this with a truth table would entail adding one attribute row for each entity to specifically record whether or not each entity has the attibute of being itself or some other entity, making the knowledge base more uniform and thereby simplify processing under the three stopping conditions mentioned above. Obviously this would make for a very large truth table with a lot of negative (and generally useless) information. If we instead represent knowledge as a matrix of questions—explicit questions about both attributes and entities—then the knowledge can be represented more tersely, more uniformly, and more usefully.

	Q2	Q3	Q4	Q5	Q6	Q7
Q1: is it alive?	0.7	0.9	0.9	0.9	0.9	0.1
Q2: can it fly?		0.9	0.8	0.9	0.2	0.1
Q3: does it lay eggs?			0.1	0.9	0.1	0.1
Q4: is it a fruitbat?				0.1	0.1	0.1
Q5: is it an eagle?					0.2	0.1
Q6: is it a tiger?						0.1
Q7: is it a rock?						

Fig. 2. A question-question matrix knowledge base

Figure 2 shows a small question-question matrix expressing more or less the same knowledge as is encoded in the truth table of Figure 1. Unlike the truth table representation, however, the matrix no longer distinguishes between entities and attributes. All knowledge is encoded uniformly as questions, making it possible to process them the same way.

A more significant difference for this knowledge base is that each cell now contains a probability instead of a truth value. Specifically, it is the probability that two questions will get the same answer, calculated simply as the number of times both questions received the same answer divided by the number of times they were both asked during an inquiry. Using probabilities opens the door for a multivalued truth system, although this was also possible with the original truth table and it is not the reason they are used here. It is not generally possible to talk of the truth relationship between two questions, but the extent to which two questions provide the same information is important when trying to decide what question to ask next. This is a complex issue and is treated more thoroughly in the next section.

For the moment, we make a couple of observations about the question-question matrix. First, we note that only the upper triangle of the matrix is needed, where the relationship of one question to all others is assessed by examining all nonempty cells in the row and column indexed by its question number. Because the assumed statistical dependency is symmetric, there is no need to use the lower triangle. Second, we observe that there is now no need to have sophisticated question generators because all linguistic knowledge is encoded explicitly as questions. When an inquiry terminates unsuccessfully, and the system needs to expand its knowledge base, it simply asks the user for one new question that would have helped it guess the target entity. Assuming a cooperative user, the input is a wellformed question that can be added to the matrix and initialised with values appropriate to the answers given over the course of the inquiry. This initialisation is also covered in the next section.

3 Question Selection and Knowledge Acquisition

Given the knowledge representation described in the previous section, we now describe how it is used to formulate a questioning strategy to satisfy the principal goal of the inquiry, and how it is updated and expanded with new knowledge.

Just as with the attribute-entity truth table, the goal of the inquirer is to select as its next question the one whose answer will best partition the search space in half. From a truth table, this is achieved by choosing the outstanding candidate question whose counts for true and false associations with eligible entities have a ratio nearest to one. When the knowledge representation is comprised of probabilities measuring the extent to which pairs of questions are expected to receive the same answer, the process is less straightforward.

The first selection function trialed was an attempt to mimic truth counting. It simply counted the number of probabilities above fifty percent and below fifty percent and selected the question with the ratio closest to one. This failed because it would not distinguish between questions whose probabilities tended to be close to fifty percent (and therefore not discriminating) and questions whose probabilities where close to either zero or one (and therefore very discriminating). The function was modified to prefer questions whose affirmative probabilities summed to a higher value, based on the assumption that the ultimate goal is to

obtain an affirmative reply to a question, but this gave the system a tendency to select crude top-level questions in preference to more refined and precise questions because the former tend to have more absolute answers than do the latter.

The key insight was to realise that the best question to ask was the one whose answer provided the most information. This is obviously what we were trying to do at the outset, but we failed to realised that the measure of information in a question is not explicitly captured in the cooccurrence statistics of "like" answers, but is given by the *mutual information* in the answers two questions are expected to elicit. Mutual information is a formal statistical property defined as

$$MI = \frac{p(A \wedge B)}{p(A) \cdot p(B)}$$

In information theory, it is more plainly described as the number of bits saved when encoding two events if the probability of one is made conditional on the other, instead of encoding both events under the assumption of independence [2]. In terms of question relevance as described here, it is a measure of the information one can expect to gain by asking a question when the answer to another question is known and assumed to be dependent. This is exactly what is needed when trying to decide which of the outstanding candidate questions is the best to pose during an inquiry. For example, if the independent probability of a positive reply to Question A is 0.5 and for Question B 0.25 then their combined information content under the assumption of independence is $-\log_2(0.5) + -\log_2(0.25) = 3$ bits. If, however, the probability of a positive reply to Question B is greater, say 0.5, given that the answer to Question A is known also to be positive, then the information content of the two questions under the assumption of a dependence is now $-\log_2(0.5) + -\log_2(0.5) = 2$ bits—a savings of one bit, which is the mutual information in the answers to the two questions.

To calculate mutual information from the knowledge base, it was necessary to modify it to store counts instead of probabilities so that the various components of the mutual information formula could be calculated as needed. And to update the knowledge base in accordance with the information embedded in the replies of the user after an inquiry, the system simply goes pairwise through all the questions asked and increments the counts for questions that received the same answer.

3.1 The Procedure

Choosing the first question to ask is a special case in that the choice must be made independently (i.e. no other answers are yet known). The first question should provide a maximum amount of disambiguation information. Ambiguity exists when mutual information is low; therefore the system selects as its first question whichever one has the lowest average mutual information with other questions; indicating that the answer will offer the greatest amount of information.

At each subsequent step, the system chooses whichever other unasked question has the least mutual information with all questions whose answer has been obtained. In this respect, the next question is whichever one provides the most new information—which is to say, the one that does the best job of raising the utility of outstanding potential questions.

The system expects the user to reply with something other than "yes" or "no" when the target entity is guessed. If it receives the response "you guessed it" then it assumes a successful line of inquiry, and the counts on question pairs are updated to reflect the proportions that they received identical answers, allowing it to pursue more effective lines of questioning on subsequent iterations.

Things are slightly less clean when the system does not know a question that will achieve successful termination. It will not stop guessing until it has exhausted all questions that have previously received at least one "yes" answer. This can become tedious in practice, but does not detract from the epistemoligcal argument.

When the system reaches a point where there are no questions left to ask, it gives up. It does this by asking the user to supply one question it could have asked to identify the target. It also asks the user what the answer to that question would be (either "yes" or "no"). Having recorded the answers to all question asked, it adds the new question to the matrix and updates the counts for affirmative and negative answers and the mutual information for all question pairs. In this manner, the system can continue to acquire new factual knowledge without limit, and improve its *understanding* of the relationships between questions.

4 Remarks

As Phil Murray (editor-in-chief of KM Briefs) recently remarked in the Knowledge Management Forum, "It is not so much knowing the answers that is important, as it is being able to formulate *correct* or effective questions"[1]. We have shown that it is possible to construct a computer system that can do just that—a system that can learn an unlimited amount of factual knowledge simply by asking appropriate propositional questions and processing their answers. The key was to represent that knowledge as the relationship between questions, rather than attempt to encode facts as explicit relationships between entities and the attributes they generally possess. We believe this to be an interesting epistemoligal view in that it partly solves the problem of what to do with exceptions. The properties that define any one entity are not so concrete as to sustain consistent truth values, even multivalued ones, because any one entity conceptually related to another of the same type class may or may not display any one particular testable attribute. The representation described in this paper allows us to define related entities in a different way—specifically, as those things for which the complete set of questions we might pose so as to test all their known attributes will on the whole elicit more or less a set of answers where the majority of answers are consistent. It would not surprise us if some readers

[1] `http://www.km-forum.org/t000013.htm`

of this report object to this claim, or have some question about the methods and results described (e.g. Question: *Does not the mutual information in the answers to a pair of question change depending on the other questions asked?* Answer: *Surprisingly, no.*) At the very least, we hope it will generate some interesting discussion for a conference on artifical intelligence.

References

1. Abney, S., Collins, M., Singhal, A.: Answer extraction. In: Proceedings of ANLP 2000, Seattle, WA (May 2000)
2. Abramson, N.: Information Theory and Coding. McGraw-Hill, New York (1963)
3. Breck, E., Light, M., Mann, G.S., Riloff, E., Anand, B.B.P., Rooth, M., Thelen, M.: Looking under the hood: Tools for diagnosing your question answering engine. In: Proceedings of the Workshop on Open-Domain Question Answering, Toulouse, France (July 2001)
4. Harabagiu, S., Pasca, M., Maiorano, S.: Experiments with open-domain textual question answering. In: Proceedings of COLING 2000, Association for Computational Linguistics/Morgan Kaufmann, San Francisco (August 2000)
5. Hintikka, J.: New foundations for a theory of questions and answers. In: Kiefer, F., Karigren, H. (eds.) Questions and Answers, pp. 159–190. KVAL, Stockholm (1983)
6. Hintikka, J.: Knowledge-seeking by questioning. In: Dancy, J., Sosa, E. (eds.) A Companion to Epistemology, pp. 241A–244A. Basil Blackwell, Oxford (1992)

A Comparison of Classification Strategies in Genetic Programming with Unbalanced Data

Urvesh Bhowan[1], Mengjie Zhang[1], and Mark Johnston[2]

[1] School of Engineering and Computer Science
[2] School of Mathematics, Statistics and Operations Research
Victoria University of Wellington, New Zealand

Abstract. Machine learning algorithms like Genetic Programming (GP) can evolve biased classifiers when data sets are unbalanced. In this paper we compare the effectiveness of two GP classification strategies. The first uses the standard (zero) class-threshold, while the second uses the "best" class-threshold determined dynamically on a solution-by-solution basis during evolution. These two strategies are evaluated using five different GP fitness across across a range of binary class imbalance problems, and the GP approaches are compared to other popular learning algorithms, namely, Naive Bayes and Support Vector Machines. Our results suggest that there is no overall difference between the two strategies, and that both strategies can evolve good solutions in binary classification when used in combination with an effective fitness function.

1 Introduction

Classification with unbalanced data represents a major challenge in the machine learning (ML) community. Data sets are unbalanced when they have an uneven representation of class examples. In binary classification, the class with the smaller number of examples is called the *minority class*, while the other class is the *majority class*. Unbalanced data sets are common; fraud detection [1], medical diagnostics [2], and image recognition [3] are only a few examples.

Genetic Programming (GP) is an evolutionary ML technique which has been successful in evolving reliable and accurate classifiers [4] [5] [6]. However, GP, like many other ML approaches, can evolve "biased" classifiers, that is, solutions with strong majority class accuracy but poor minority class accuracy, when data sets are unbalanced [2] [5] [6]. As the minority class often represents the main class in many problems, accurately classifying examples from this class can be *at least* as important, and in some scenarios more important, than accurately classifying examples from the majority class.

This performance bias is typically due to traditional training criteria being influenced by the larger majority class [1] [2] [7]. In GP, the standard fitness function for classification (overall classification accuracy) is known to evolve biased solutions compared to improved fitness functions that are more sensitive to the uneven representation of examples in each class [7]. These improved functions focus on evolving classifiers with good accuracy on both classes; examples include

J. Li (Ed.): AI 2010, LNAI 6464, pp. 243–252, 2010.

approximating the area under the Receiver Operating Characteristics (ROC) curve (known as the AUC) in fitness [5] [6], or using fixed misclassification costs for class examples to boost classification rates [2] [7].

In binary classification, GP classifiers are usually represented as mathematical expressions, where the numeric program output is mapped to two class labels using a fixed class-threshold (zero) [5] [7]. However, recent work using GP for multi-class classification has shown that a more flexible strategy using a *dynamic* class-threshold, where the decision boundary is determined on a solution-by-solution basis, can lead to better performance of evolved solutions compared to the standard zero-threshold approach [8]. This paper investigates whether this dynamic-threshold can also lead to improvements in the performance of evolved solutions in classification with unbalanced data, compared to the standard zero-threshold approach. We compare the effectiveness of these two classification strategies using five different GP fitness functions across a range of unbalanced data sets. The different GP fitness functions are included in this comparison to allow for both a fair comparison of the two strategies over different GP systems, and a thorough evaluation of these fitness functions when data is unbalanced. The GP fitness functions include the standard approach for classification (overall accuracy), and four improved functions for class imbalance. The GP approach is also compared to Naive Bayes and Support Vector Machines.

The rest of this paper is organised as follows. Section 2 outlines the GP framework and two classification strategies. Section 3 describes the GP fitness functions. Section 4 outlines the experimental setup and unbalanced data sets. Section 5 presents the full experimental results and analysis. Section 6 concludes this paper and gives directions for future work.

2 GP Framework and Classification Strategies

The genetic programs are represented as (tree-based) mathematical expressions [4]. We use features and randomly generated floating-point numbers in the terminal set, and a function set comprising of the four standard arithmetic operators, $+,-,\times$ and %, and conditional operator, `if`. The $+,-$ and $\times$ operators have their usual meanings (addition, subtraction and multiplication) while % is *protected* division (usual division except that a divide by zero gives a result of zero). These four operators take two arguments and return one. The `if` operator takes three arguments and returns the second if the first is negative, otherwise the third is returned. For example, the expression $(F_1 \times F_2 - 0.5)$ can be represented by genetic program `(- (× F1 F2) 0.5)`, where the arithmetic operators are the functions, and the variables and constants are the terminals.

Two Classification Strategies

As a mathematical expression, a genetic program classifier computes a single output value (floating-point number) for a particular data example that must be classified. A common technique to translate this number into binary class labels uses the *zero-threshold* classification strategy: an example will be assigned

to the majority class if the classifier output is negative, otherwise it will be assigned to the minority class [5] [7].

However, in difficult problems the fixed zero-threshold approach can unnecessarily place additional constraints on solutions during evolution. Classifiers are required not only to separate the predictions for each class, but also ensure that minority predictions are non-negative and majority predictions are negative. For example, a GP solution could have good class separability (i.e., little overlap in class predictions) but poor classification performance using the zero class-threshold (Figure 1a). Further evolving the solution to "shift" its predictions relative to the zero class-threshold could negatively impact on separability (Figure 1b).

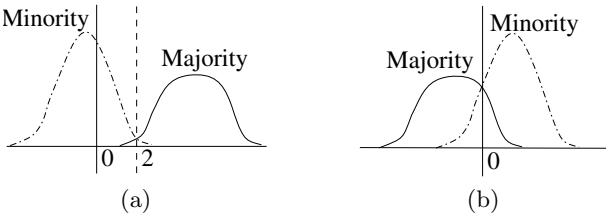


Fig. 1. Example distributions for the minority and majority classes for two classifiers (x-axis corresponds to the genetic program output). Solid line is zero class-threshold, dashed line in (a) is better class-threshold.

A more flexible classification strategy would use a *dynamic* class-threshold to find the best decision boundary relative to the class predictions on a solution-by-solution basis, e.g., the dashed line in Figure 1a. To automatically find the optimal class-threshold during evolution, a probabilistic classification strategy is used where the outputs of genetic programs were modelled using two Gaussian distributions, one for each class; the normal probability density function (ϕ) of the class distributions is used to determine the class label of a given example [8].

The two class distributions correspond to the outputs of the genetic program classifier when evaluated on all training examples from the two classes (Figure 1 shows example distributions for two classifiers). To predict the class of an unseen example, two ϕ values are calculated (one for each class distribution), and the class with the higher ϕ value is taken as the class of that particular example.

In Eq. (1), μ_c and σ_c are the mean and standard deviation of the distributions for class c (minority or majority class), respectively, N_c is the number of examples in class c, P_i is the program output on unseen example i, and P_j is the program output on training example j:

$$\phi(\mu_c, \sigma_c, P_i) = \frac{exp\left(\frac{-(P_i-\mu_c)^2}{2\sigma_c^2}\right)}{\sigma_c\sqrt{2\pi}} \tag{1}$$

where

$$\mu_c = \frac{\sum_{j\in c} P_j}{N_c} \quad \text{and} \quad \sigma_c = \sqrt{\frac{1}{N_c}\sum_{j\in c}(P_j - \mu_c)^2}$$

3 GP Fitness Functions

The classification strategy determines *how* class labels are assigned to examples in the data set. The fitness function is different; this measures the classification accuracy of a solution by comparing the predicted class labels to the actual class labels in the training set. The behaviour of the two classification strategies is compared using five different GP fitness functions, outlined below.

Fitness Function 1: The function *Std* (Eq. 2) corresponds to standard GP fitness for classification, that is, the overall classification accuracy of a solution on the training examples. Assuming the minority and majority classes are the *positive* and *negative* classes, respectively, *Std* can be defined using the confusion matrix in Table 1.

Table 1. Confusion matrix for a two-class classification problem

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

$$Std = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

Fitness Function 2: Research has shown that fitness function *Std* can favour the evolution of classifiers biased toward the majority class [5] [6] [7]. When data sets are unbalanced, a solution with relatively high overall classification accuracy can have poor minority class accuracy due to the influence of the majority class. The fitness function *Ave* (Eq. 3) uses the average classification accuracy of each class to avoid this pitfall, where solutions with good accuracy on both classes are favoured with better fitness, compared to biased solutions with strong accuracy on only one class.

$$Ave = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{3}$$

AUC Approximation in Fitness

The three functions presented below explore effective approximations to the AUC in fitness. The AUC is a useful, non-biased performance measure of classifier *separability* but is often impractical to use directly in the fitness function during evolution due to the computational overhead required to construct an ROC curve [9].

Fitness Function 3: The function *Wmw* (Eq. 4) uses the well-known equivalent estimator of the AUC, the Wilcoxon-Mann-Whitney (WMW) statistic in fitness [6] [9] [10]. In Eq. (4), P_i and P_j represent the output of a genetic program when evaluated on an example from the minority class (min) and majority class (maj), respectively, where indicator function I_s depends on the classification

strategy s. For the zero class-threshold, I_{ZT} (Eq. 5) was used; for the dynamic class-threshold, I_{DT} (Eq. 6) was used.

$$Wmw = \frac{\sum_{i \in min} \sum_{j \in maj} I_s(P_i, P_j)}{N_{min} \times N_{maj}} \tag{4}$$

Eq. (5) measures the ordering of minority class predictions to both majority class predictions and the fixed zero-threshold decision boundary. Eq. (6) measures only the ordering of minority to majority class predictions, depending on which class distribution is the left-most distribution (distribution with the smallest mean).

$$I_{ZT}(P_i, P_j) = 1 \text{ if } P_i > 0 \text{ and } P_i > P_j, \text{ 0 otherwise.} \tag{5}$$

$$I_{DT}(P_i, P_j) = \begin{cases} 1 & \text{if } (\mu_{min} \geq \mu_{maj} \text{ and } P_i > P_j) \\ & \text{or } (\mu_{min} < \mu_{maj} \text{ and } P_i < P_j), \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Fitness Function 4: The function *Corr* (Eq. 7) is a relatively new fitness function which makes novel use of the *correlation ratio* to approximate the AUC [7]. The correlation ratio measures the linear dispersal between two populations of data, where the higher the dispersal, the better the class separability. In Eq. (7), P_{ci} represents the genetic program output when evaluated on the i^{th} example belonging to class c, M is the number of classes, and N_c is the number of examples for class c. Function *Corr* will return values ranging between 0 (poor separability) and 1 (good separability):

$$Corr = \sqrt{\frac{\sum_{c=1}^{M} N_c(\mu_c - \bar{\mu})^2}{\sum_{c=1}^{M} \sum_{i=1}^{N_c} (P_{ci} - \bar{\mu})^2}} \tag{7}$$

where μ_c is the mean of classifier outputs for class c, and $\bar{\mu}$ is the overall mean:

$$\mu_c = \frac{\sum_{i=1}^{N_c} P_{c,i}}{N_c} \text{ and } \bar{\mu} = \frac{\sum_{c=1}^{M} N_c \mu_c}{N}$$

Function *Corr* only measures class separability; to evolve solutions that adhere to the zero class-threshold strategy, *Corr* must be adapted to reward solutions with majority and minority class predictions that are negative and non-negative, respectively. This was enforced using function $Adjusted_{ZT}$ (Eq. 8), which doubles the original fitness value if solutions adhere to the zero class-threshold. In Eq. 8, fit corresponds to the original fitness value returned by a given fitness function, in this case *Corr* (Eq. 7).

$$Adjusted_{ZT} = fit \times \begin{cases} 2 & \text{if } \mu_{maj} < 0 \text{ and } \mu_{min} \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Fitness Function 5: The function *Dist* (Eq. 9), measures the distance between the majority and minority class distributions for a given classifier [8]. Eq. (9) returns the point equi-distant from the means of two distributions. In the worst case

where the means of both class distributions are the same (poor separability), this distance will be 0. Conversely when there is no overlap between the two class distributions (good separability), this distance will be large (go to $+\infty$). Similar to $Corr$, this function was also adapted for the zero-threshold classification strategy using function $Adjusted_{ZT}$ (Eq. 8), where fit is the value returned from $Dist$.

$$Dist = \frac{|\mu_{min} - \mu_{maj}|}{\sigma_{min} + \sigma_{maj}} \tag{9}$$

4 Experimental Parameters, Setup and Data Sets

The ramped half-and-half method was used for generating programs in the initial population and for the mutation operator [4]. The population size was 500, maximum program depth was 8 (to restrict very large programs in the population), and crossover, mutation and elitism rates were 60%, 35%, and 5% respectively. The evolution ran for 50 generations or until a solution with optimal fitness was found. Half of the data set was randomly chosen as the training set and the other half as the test set, where the examples in each set were different for every experiment. Both sets preserved the original class imbalance ratio.

The AUC of the evolved solutions (on the test set) was used as the main performance evaluation metric in our analysis, as this is known to be an accurate single-figure measure classification ability [1] [9] [10]. The AUC represents a solution's classification performance across varying true-positive and false-positive thresholds and is insensitive to the learning bias (unlike standard overall accuracy) [9]. To compute the AUC for a given solution, an ROC curve was constructed and the well-known trapezoidal approximation technique was used to calculate the area under the curve (i.e., the sum of the areas of the individual trapezoids fitted under each ROC point [9]).

To compare the classification results of the GP approaches with other popular machine learning approaches, we also trained a Naive Bayes and a Support Vector Machine classifier on the unbalanced data sets (using WEKA [11]). A sequential minimal optimisation algorithm was used to train the SVM, with an RBF kernel and gamma value of 10 (giving the best results from 0.1, 1 and 10).

Unbalanced Data Sets

Five benchmark binary classification problems were used in the experiments, taken from the *UCI Repository of Machine Learning Databases* [12], and the Intelligent Systems Lab at the University of Amsterdam [3].

Ionosphere (Ion). This data set contains 351 recorded radar signals collected using 17 high-frequency antennas targeting free electrons in the ionosphere. There are 126 instances of "good" signals (35.8%) and 225 "bad" signals (64.2%), a class imbalance ratio of roughly 1:2. Signals were processed using an auto-correlation function returning two attributes per pulse, giving 34 real-number features (F_1–F_{34}) [12]. There are no missing attributes.

Spect Heart (Spt). This data set contains 267 records derived from cardiac Single Proton Emmision Computed Tomography (Spect) images. There are 55

"abnormal" records (20.6%) and 212 "normal" records (79.4%), an imbalance ratio of approximately 1:4. The images contain 22 binary features (F_1–F_{22}) that make up the attributes for each instance [12]. There are no missing attributes.

Yeast (Yst_1 and Yst_2). This data set contains 1482 instances of protein localisation sites in yeast cells, with eight amino-acid sequences as numeric features (F_1–F_8) [12]. This is a multi-class classification problem where each class has a different degree of class imbalance. For our purposes, this problem was decomposed into *many* binary classification tasks with only one "main" (minority) class and everything else as the majority class. Two "main" classes were used: Yst_1 had 244 minority class examples (16%), an imbalance ratio of 1:6; and Yst_2 had 163 minority class examples (11%), an imbalance ratio of 1:9.

Pedestrian images (Ped). This data set contains 24,800 (PGM-format) cut-outs of 4,800 pedestrian (19.4%) and 20,000 (80.6%) non-pedestrian (background) images, an imbalance ratio of approximately 1:4 [3]. These images are 19×36 pixels in size. 22 low-level pixel statistical features, F_1–F_{22}, corresponding to the mean and variance of pixel values around certain local regions in the image were used as image features. Details can be seen in [7].

5 Experimental Results and Analysis

Table 2 shows the average (with standard deviation) and best AUC of evolved solutions using the zero and dynamic class-thresholds with the different fitness functions, over 50 runs. These results also show which classification strategy was *significantly better* in terms of AUC performances using a particular fitness function (denoted by symbol ▲), according to the 95% confidence interval of the AUC differences[1] over 50 runs. Note that absence of symbol ▲ means that there was no significant difference between the two classification strategies. Table 2 also shows the average GP training times in seconds (s) or minutes (m), as well as the AUC and run times (in parenthesis) using Naive Bayes (NB) and Support Vector Machines (SVM) on the problems. The best AUC reached by any classifier is underlined for each problem. Note that no GP run found a solution with optimal fitness on the training set for any problem.

Table 2 shows, that not surprisingly, the standard GP fitness function for classification (*Std*) generally gave poorer AUC performances compared to the improved GP fitness functions (*Wmw*, *Dist*, *Corr*, and *Ave*) using either classification strategy. Closer inspection of these results showed that the solutions evolved using *Std* were usually biased toward the majority class (low minority class accuracy). On the other hand, solutions evolved using the improved GP fitness functions had higher minority class accuracies and a better balance in class performances (class accuracies omitted due to space constraints).

Overall, using the standard GP fitness function (*Std*), Table 2 shows that there is no major difference between the two classification strategies. The zero-threshold (ZT) strategy showed significantly better results in two tasks (Ion and Spt), while

[1] Calculated using the "common random numbers" technique on a run-by-run basis (i^{th} run for both strategies used the same random start seed and initial population).

Table 2. Average (± standard deviation) and best AUC, and average run time using two GP classification strategies and fitness functions (50 runs). Symbol ▲ denotes a significantly better AUC performance (95% confidence interval). AUC and run time (in parenthesis) using Naive Bayes (NB) and Support Vector Machines also included. Best performance using any approach is underlined for each problem.

Data Set	GP Fitness Function	GP Zero-Threshold AUC Average	GP Zero-Threshold AUC Best	GP Zero-Threshold Time	GP Dynamic-Threshold AUC Average	GP Dynamic-Threshold AUC Best	GP Dynamic-Threshold Time	NB	SVM
Ion	Std	▲ 0.81 ± 0.05	0.89	3.1s	0.74 ± 0.11	0.91	3.3s	0.91 (0.02s)	0.93 (0.08s)
	Ave	▲ 0.81 ± 0.05	0.91	3.0s	0.77 ± 0.10	0.92	3.0s		
	Wmw	0.82 ± 0.05	0.93	3.6s	▲ 0.86 ± 0.04	0.93	3.6s		
	Corr	0.89 ± 0.04	0.94	2.5s	0.89 ± 0.03	0.96	2.6s		
	Dist	▲ 0.88 ± 0.04	0.96	1.4s	0.86 ± 0.04	0.94	1.4s		
Spt	Std	▲ 0.72 ± 0.07	0.81	2.4s	0.64 ± 0.09	0.80	3.5s	0.83 (0.04s)	0.68 (0.2s)
	Ave	▲ 0.71 ± 0.05	0.81	2.5s	0.68 ± 0.10	0.82	2.9s		
	Wmw	0.74 ± 0.06	0.85	3.0s	0.76 ± 0.05	0.85	3.0s		
	Corr	0.74 ± 0.05	0.84	2.5s	0.73 ± 0.05	0.81	2.4s		
	Dist	▲ 0.74 ± 0.05	0.83	1.3s	0.70 ± 0.07	0.82	1.4s		
Ped	Std	0.82 ± 0.07	0.91	5.6m	▲ 0.85 ± 0.10	0.92	5.6m	0.92 (0.13s)	0.93 (3.8m)
	Ave	0.86 ± 0.04	0.90	5.4m	0.88 ± 0.09	0.92	5.0m		
	Wmw	0.89 ± 0.01	0.92	64.8m	▲ 0.93 ± 0.01	0.94	50.1m		
	Corr	0.89 ± 0.03	0.92	4.6m	▲ 0.91 ± 0.01	0.92	4.3m		
	Dist	0.89 ± 0.01	0.92	2.5m	▲ 0.91 ± 0.03	0.93	2.4m		
Yst_1	Std	0.75 ± 0.09	0.85	14.1s	0.73 ± 0.11	0.86	15.2s	0.83 (0.03s)	0.71 (1.2s)
	Ave	0.79 ± 0.04	0.86	12.9s	▲ 0.82 ± 0.03	0.87	13.1s		
	Wmw	0.82 ± 0.02	0.87	23.9s	▲ 0.83 ± 0.02	0.87	19.9s		
	Corr	0.81 ± 0.03	0.87	12.4s	▲ 0.83 ± 0.02	0.86	11.7s		
	Dist	0.82 ± 0.02	0.87	6.0s	0.82 ± 0.03	0.88	6.0s		
Yst_2	Std	0.87 ± 0.09	0.97	13.3s	0.90 ± 0.11	0.98	14.4s	0.95 (0.02s)	0.85 (1.4s)
	Ave	0.93 ± 0.04	0.97	12.6s	▲ 0.95 ± 0.03	0.98	14.1s		
	Wmw	0.95 ± 0.02	0.98	16.6s	0.95 ± 0.02	0.98	16.1s		
	Corr	0.95 ± 0.02	0.98	11.3s	0.95 ± 0.03	0.98	10.2s		
	Dist	0.95 ± 0.03	0.98	5.5s	0.95 ± 0.03	0.98	5.5s		

the dynamic-threshold (DT) strategy was better in one task (Ped); there was no significant difference between the two strategies in the remaining two tasks. This behaviour is similar using the improved fitness functions *Ave* and *Dist*, where the ZT strategy was better in some tasks (Ion and Spt) while DT was better in others (Yst_1 and Yst_2 for *Ave*, and Ped for *Dist*). However, the DT strategy did lead to small (but statistically significant) improvements using two improved fitness functions, *Wmw* and *Corr*, in some problems. *Wmw* was better in three tasks and *Corr* was better in two (no statistical difference in the other tasks).

These results suggest that while the DT strategy is better than ZT for multi-class classification, this is not the case for problems with two classes only. In binary classification, either GP classification strategy can lead to good solutions being evolved when used with a good fitness function, as the evolved GP classifiers are generally able to "shift" their class predictions relative to the zero

Table 3. First, second and third place AUC rankings for the GP fitness functions. The highest possible rank is 250 (same ranked position in 50 experiments over five tasks).

Fitness	GP ZT			GP DT		
Function	1st	2nd	3rd	1st	2nd	3rd
Std	13	30	37	10	23	30
Ave	18	30	48	21	44	58
Wmw	54	56	64	120	47	48
Corr	77	71	59	57	70	54
Dist	89	64	42	42	66	62

class-threshold during evolution (e.g., by tweaking the mathematical expressions representing the classifiers), without negatively impacting class separability. Most other ML techniques cannot easily achieve this during training.

Comparing the performance of the GP approaches to NB and SVM, the best-evolved GP solution always achieved the highest AUC performance by any classifier on these problems. This shows that GP in combination with a good fitness function is capable of evolving high-AUC classifiers that outperform both NB and SVM on the tasks. On average, GP had similar performance to NB when the level of class imbalance in a problem was high (e.g., more than 80% majority class such as Ped and Yst). For highly unbalanced tasks (such as Yst) SVM showed the lowest AUC. When the class distributions were more balanced (such as Ion), NB and SVM showed similarly good results.

Effectiveness of Different Fitness Functions

To determine which GP fitness functions were most effective, the AUC for each task was *ranked* by fitness function into first, second and third places, on a run-by-run basis (where the i^{th} run of every method shared the same starting seed). The number of first, second and third place rankings using a particular fitness function was then counted over all tasks, as shown in Table 3.

Table 3 shows that using the ZT classification strategy, functions $Dist$ and $Corr$ achieved the best AUC results; these were ranked in first place most often and also had the highest number of total top-three rankings. These functions out-ranked and incurred considerably shorter training times (Table 2) than the well-established Wmw function, suggesting that $Dist$ and $Corr$ are fast and effective measures to evolve solutions with high AUC using the ZT classification strategy on these tasks. On the other hand, Wmw clearly out-ranked all other functions using the DT classification strategy, suggesting that Wmw was most effective when used in combination with DT strategy. For both classification strategies the standard GP fitness, Std, produced the worst ranks.

6 Conclusions

The main goals of this paper were to investigate whether a more flexible dynamic-threshold classification strategy improved the performance of evolved solutions compared to the standard zero-threshold strategy, and compare the effectiveness of five different fitness functions across a range of unbalanced data sets.

Our results show that for the standard GP fitness function and two improved fitness functions for classification, there is no overall difference between the two strategies. For the remaining two functions (the correlation ratio and WMW statistic in fitness), the dynamic class-threshold strategy did lead to small improvements over the zero-threshold strategy on some problems. In terms of the most effective fitness functions on the tasks, two new functions using the correlation ratio and the distance between the class distributions in fitness, achieved the best AUC rankings (using the zero-threshold strategy), outperforming the well-known WMW-based function. Using the dynamic class-threshold, the WMW-based function performed the best suggesting that this fitness function better exploited this classification strategy in the evolutionary phase.

For future work we plan to develop new GP fitness functions to improve the performance of the evolved solutions on these and other unbalanced data sets.

References

1. Fawcett, T., Provost, F.: Adaptive fraud detection. Data Mining and Knowledge Discovery 1, 291–316 (1997)
2. Holmes, J.H.: Differential negative reinforcement improves classifier system learning rate in two-class problems with unequal base rates. In: Koza, J.R., Banzhaf, W., Chellapilla, K., et al. (eds.) Genetic Programming 1998: Proceedings of the Third Annual Conference, pp. 635–644. Morgan Kaufmann, San Francisco (1998)
3. Munder, S., Gavrila, D.: An experimental study on pedestrian classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1863–1868 (2006)
4. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
5. Winkler, S., Affenzeller, M., Wagner, S.: Advanced genetic programming based machine learning. Journal of Mathematical Modelling and Algorithms 6(3), 455–480 (2007)
6. Doucette, J., Heywood, M.I.: GP classification under imbalanced data sets: Active sub-sampling and AUC approximation. In: O'Neill, M., Vanneschi, L., Gustafson, S., Esparcia Alcázar, A.I., De Falco, I., Della Cioppa, A., Tarantino, E. (eds.) EuroGP 2008. LNCS, vol. 4971, pp. 266–277. Springer, Heidelberg (2008)
7. Bhowan, U., Johnston, M., Zhang, M.: Genetic programming for image classification with unbalanced data (ivcnz). In: Proceedings of 24th International Conference on Image and Vision Computing, Wellington, New Zealand, pp. 316–321. IEEE Press, Los Alamitos (2009)
8. Zhang, M., Smart, W.: Using Gaussian distribution to construct fitness functions in genetic programming for multiclass object classification. Pattern Recognition Letters 27(11), 1266–1274 (2006)
9. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30, 1145–1159 (1997)
10. Yan, L., Dodier, R., Mozer, M.C., Wolniewicz, R.: Optimizing classifier performance via the Wilcoxon-Mann-Whitney statistic. In: Proceedings of The Twentieth International Conference on Machine Learning (ICML 2003), pp. 848–855 (2003)
11. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations 11(1) (2009)
12. Asuncion, A., Newman, D.: UCI Machine Learning Repository. University of California, Irvine (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html

An Encoding Scheme for Cooperative Coevolutionary Feedforward Neural Networks

Rohitash Chandra[1,2], Marcus Frean[1], and Mengjie Zhang[1]

[1] School of Engineering and Computing Science,
Victoria Universty of Wellington, P.O. Box 600,
Wellington, New Zealand
{rohitash.chandra,marcus.frean,mengjie.zhang}@ecs.vuw.ac.nz
[2] Department of Computing Science and Information Systems,
Fiji National University, Suva, Fiji

Abstract. The cooperative coevolution paradigm decomposes a large problem into a set of subcomponents and solves them independently in order to collectively solve the large problem. This work introduces a novel encoding scheme for building subcomponents based on functional properties of a neuron. The encoding scheme is used for training feedforward neural networks. The results show that the proposed encoding scheme achieves better performance when compared to its previous counterparts.

Keywords: Feedforward neural networks, cooperative coevolution, evolutionary algorithms.

1 Introduction

Cooperative coevolution (CC) decomposes a large problem into subcomponents and solves them independently in order to collectively solve the large problem. The subcomponents are implemented as subpopulations. The subpopulations in the cooperative coevolution framework are evolved separately and the cooperation only takes place for fitness evaluation for the respective individuals in each subpopulation. The way a subcomponent is designed is dependant on the problem. CC has shown promising results in neuro-evolution and general function optimisation problems [1,2,3,4]. A major concern in CC based neuro-evolution is how the subcomponents are designed from the respective network topology.

In recent years, two major subcomponent design methodologies have been proposed. These include subcomponent design on the *neuron level* and *synapse level*. The neuron level subcomponent design uses each neuron in the hidden layer as the main reference point for the respective subcomponent. Each subcomponent consists of the incoming and outgoing connections. The cooperative coevolution model for evolving artificial neural networks (COVNET) [5] and multi-objective cooperative networks (MOBNET) [6] build subcomponents by encoding input and output connections to the respective hidden neuron. They have been used for training feedforward network architectures. This encoding

J. Li (Ed.): AI 2010, LNAI 6464, pp. 253–262, 2010.

scheme is similar to that of *enforced subpopulations* (ESP) for training recurrent neural networks [7,8], which has been applied to pole balancing problems.

In the synapse level encoding, each weight or link in the network forms a subcomponent of the CC framework. The cooperatively coevolved synapse neuroevolution (CoSyNE) algorithm was used for training feedforward and recurrent networks on pole balancing problems [9]. In this encoding scheme, a subcomponent represents a single interconnection which is either the weight or bias in the network. Therefore, the number of subpopulations depends on the number of weights and biases.

This paper examines the neural network encoding schemes from literature for pattern recognition problems. It compares their performance on benchmark datasets and proposes a new encoding scheme that is similar to the neuron level. The proposed subcomponent design is called neuron-based subpopulation (NSP), and is based on the functional properties of a neuron. The aim of NSP is to train the network similarly to the other encoding schemes – but at the same time – reduce the training time in terms of number of function evaluations. We use four benchmark problems from the UCI machine learning repository [10] for evaluating the performance of the NSP in comparison with those from literature. We investigate on the optimal depth of search required for the respective encoding schemes in the CC frameworks for training feedforward networks. Our goal is to maintain high levels of accuracy in terms convergence while achieving the fewest fitness function evaluations.

The rest of the paper is organised as follows. Section 2 presents the encoding scheme in the proposed neuron based subpopulation. Section 3 presents the results and section 4 concludes the work with a discussion on future work.

2 The Neuron Based Subpopulation

2.1 Preliminaries

Note that the ESP was used for training recurrent neural networks and therefore recurrent connections were encoded in the subcomponents. This is not present in either of COVNET or MOBNET. We use the general idea behind ESP and build a similar encoding scheme which encodes the incoming and outgoing connections relative to a hidden neuron as a subcomponent. This encoding scheme is used for comparing with the performance of the NSP method proposed in this work. Figure 1 shows the schematic of the interconnected input and output links to a hidden neuron. It is assumed that the network has one hidden layer only. The number of hidden neurons is equal to the the number of subcomponents. In this encoding scheme, all individual subpopulations have the same size for the entire framework.

Henceforth, we refer to the encoding schemes used in COVNET, MOBNET, ESP due to their similarities as "CME" by taking the first letters from each abbreviation. ESP and COVNET do not have fixed structure in terms of hidden neurons. They adapt the number of hidden neurons in the evolutionary process. However, for comparison of encoding schemes in this work, the CME structure is

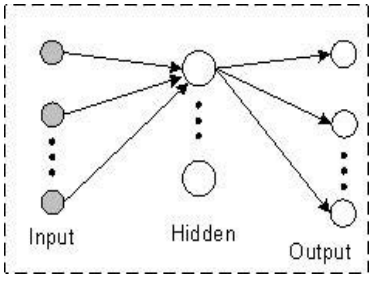


Fig. 1. The CME encoding scheme is summarised from [8,5,6] is used for comparison in our experiments

fixed. Note that the proposed NSP scheme also has a fixed structure. The advantage of CoSyNE over its previous counterparts is that it achieves a higher level of modularity with a greater number of subpopulations. This is costly as the number of function evaluations increases with a greater number of subcomponents in the initialisation stage in comparison to CME.

2.2 The New Encoding Scheme: Neuron Based Subpopulation (NSP)

There is a need to balance the level of modularity used in the CC paradigms for training neural networks. It has been outlined that the level of modularity used CoSyNE produces a larger number of function evaluations in the initialisation stage of the CC framework.

The CME on the other hand, has lower level of modularity, meaning lower number of function evaluations in the initialisation phase. Although this is an advantage, it fails to deliver the same level of performance in terms of fitness evaluations when compared to CoSyNE [9]. An efficient subcomponent encoding scheme should aim to decrease the number of function evaluations at the initialisation stage when compared to CoSyNE, while at the same time, it should give the same or better optimisation performance.

A single subcomponent in CME encodes the incoming and outgoing connections in reference to a hidden neuron. The neuron based subpopulation (NSP) breaks down this encodings scheme into a lower level and achieves a higher level of modularity. Each subcomponent in the NSP consists of incoming connections associated with neurons in the hidden and output layers. The NSP is motivated by the properties of a neuron as its output is dependant on the weighted sum of incoming weight links only. The output of a neuron does not depend on the outgoing connection links in this computation unlike CME. NSP uses one subpopulation for each neuron. Therefore, each subpopulation for a layer is composed of the following:

1. Hidden layer subpopulations: weight-links from each neuron in the $hidden_j$ layer connected to all $input_i$ neurons and the bias of $hidden_j$.
2. Output layer subpopulations: weight-links from each neuron in the $output_k$ layer connected to all $hidden_j$ neurons and the bias of $output_k$

Figure 2 shows a detailed diagram of the NSP encoding scheme. The general CC framework for NSP in training feedforward networks is summarised in Algorithm 1. Each neuron in the hidden and output layer acts as a reference point to its subcomponents given as subpopulations.

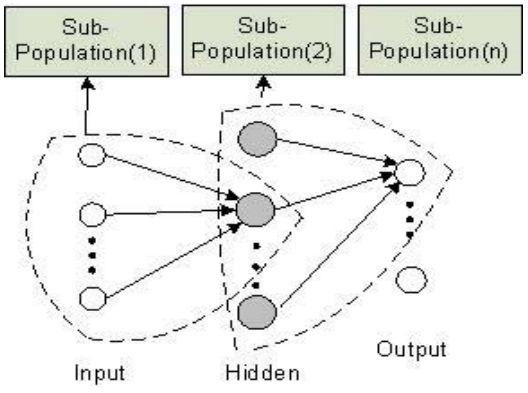


Fig. 2. The NSP encoding scheme. The same encoding scheme is used in the rest of the neurons in the hidden and output layer.

In Algorithm 1, the network evolution problem is decomposed into k subcomponents where k is equal to the number of hidden neurons, plus the number of output neurons. Each subpopulation contains all the weight links from the previous layer connecting to a particular neuron. A *Cycle* is completed when all the subpopulations are evolved for a fixed number of function evaluations or generations. At the end of each Cycle, the best components of each subpopulation are cooperatively evaluated. The algorithm halts if the termination condition is satisfied. The termination condition is when the network correctly classifies a given percentage of the training data or when the maximum training time is reached.

A major concern in the general paradigm is the cooperative evaluation of each subcomponent in every subpopulation, especially in the initialisation and evaluation phases shown in Step 3 of Algorithm 1. This is simply done by selecting arbitrary individuals from each subpopulation in the initialisation phase. In order to evaluate the ith individual of the kth subcomponent, the best individuals from the rest of the subpopulations would be combined with the chosen individual and cooperatively evaluated. However, the fitness of the individual is not divided among the rest of the subpopulations [3].

Algorithm 1. The NSP CC Framework for Training FNN

Step 1) Decompose the problem into k subcomponents according to the number of Hidden and Output neurons. k is the number of hidden neurons.
Step 2) Encode each subcomponent in a subpopulation in the following order:

1. Hidden layer subpopulations
2. Output layer subpopulations

Step 3) Initialise and cooperatively evaluate each subpopulation
for each Cycle until termination **do**
  for each Subpopulation **do**
    for n Generations **do**
      i) Select and create new offspring
      ii) Cooperatives Evaluate the new offspring
      iii) Add new offspring's to the subpopulation
    end for
  end for
end for
Get the best solution

Cooperative evaluation is done by combining or concatenating the chosen individual from a subpopulation k with the best individuals from the rest of the subpopulations. Note that the position of the particular subpopulation in the chromosome is retained during concatenation. The final chromosome is encoded into the network and the training data is used to calculate the network error, which becomes the inverse of the fitness value. The goal of the evolutionary process is to increase the fitness which tends to decreases the network error. In this way, the fitness of each subcomponent in the network is evaluated until the Cycle is completed.

Other CC frameworks such as CME and CoSyNE evaluate the fitness of each individual in n trial runs and take the average or best fitness which is done only in the initialisation phase. We have discussed that MOBNET addresses the fitness assignment problem through multi-objective optimisation methods. The NSP framework will use the fitness evaluation given in [3] for the initialisation of subpopulations since this initialisation methodology uses smaller function evaluations than doing n trial runs. In order to show a fair comparison with NSP, the CoSyNE and CME, we will use the same method for subpopulation initialisation and evaluation during evolution.

We note that most CC paradigms have used older evolutionary algorithms in their subpopulations which include ESP, COVNET, MOBNET and CoSyNE. Therefore, this work employs an efficient evolutionary algorithm (G3-PCX) [11] in the subpopulations of the proposed CC framework. This evolutionary algorithm has shown good performance in training feedforward neural networks [12].

3 Simulation and Analysis

This section presents a experimental study of the NSP and compares it with CoSyNE and CME discussed in the previous sections. The G3-PCX [11] evolutionary algorithm is used in standard neuro-evolution and all the respective CC frameworks.

The G3-PCX algorithm uses a population size of 100, 2 offspring and 3 parents (of which 1 is the current best) for all the respective CC frameworks. This set-up has been used in [11] and has shown good results for general optimisation problems. The subpopulations are initialised with random real numbers in the range of [-5, 5] in all experiments.

3.1 Real-World Problems and Neural Network Configuration

The datasets obtained from the UCI machine learning repository [10] included **Iris, Wine, Cleveland Heart Disease** and **Wisconsin breast Cancer**.

Table 1 shows the neural network configuration and dataset details used for all the experiments. The data is split into training and testing sets where 70 percent of the data is used for training and the remaining 30 percent for testing. The maximum training time given by the number of function evaluations in all the problems was fixed as 100000. The table also shows the minimum training performance required for each problem. Note that only in the heart disease classification problem, the neural networks is trained until is reaches at-least 88 percent classification performance on the training data. This value was determined in sample runs as it was seen that reaching a better classification performance was difficult for this problem. In all other problems, the minimum classification performance on the training data is 95 percent. The network topology configuration for each problem is also given in Table 1. The number of hidden units was chosen from trial experiments.

We did not do cross-validation for each dataset with the respective CC paradigm as our goal was not to test on the generalisation given by the network architecture, but to observe the training performance of each paradigm on the same training and testing data split.

3.2 Number of Generations in Subpopulation

Each subpopulation is evolved for a fixed number of generations in NSP CC framework shown in Algorithm 1. Note that all subpopulations are meant to evolve for the same number of n generations which is fixed beforehand.

Table 1. Dataset information and neural network Configuration

Domain	Cases	Class	Feat.	Input	Out.	Hid.	Min. Train (%)
Wine	178	3	13	13	3	4	95
Iris	150	3	4	4	3	4	95
Heart	303	2	13	13	1	7	88
Breast Cancer	699	2	9	9	1	5	95

The results are given in Tables 2 and 3 which report the number of function evaluations (FuncEval) and the number of successful runs (Success) out of 30 runs. The depth of search (Depth) given by the number of generations used for all subpopulations and the performance on the test set in percentage (Test Error) are also shown. The optimal run is determined by the least number of function evaluations and greater number of successful runs. The values in the subscript shows the standard deviation. Note that results from the unsuccessful runs are not included in the mean and the standard deviation. Best results are shown in bold.

The results show that the optimal performance is given when the CC framework uses a lower depth search from 1 to 16 generations for NSP, CME and CoSyNE. The optimal depth is shown in bold respective entries of Tables 2 and 3 which reports the least number of function evaluations with the best number of successful runs.

Table 2. Performance in Iris and wine classification problems

		Iris					**Wine**				
Method	Depth	FuncEval		Test Error		Success	FuncEval		Test Error		Success
NSP	1	**13056**	**5970**	95.08	1.08	30	**13644**	**3063**	94.58	1.28	30
	6	15598	8058	94.47	1.34	30	14455	5504	94.67	0.97	30
	11	**12513**	**3635**	94.91	1.24	30	**13003**	**3483**	92.41	1.70	30
	16	13239	3587	93.51	1.28	30	13232	3116	93.67	1.47	30
	21	13327	5395	94.55	0.97	29	14845	4076	93.41	1.43	30
	26	15908	8103	94.74	1.06	30	13797	4536	94.41	1.38	30
CME	1	**35688**	**12648**	95.17	1.19	12	**17856**	**8306**	94.64	1.12	28
	6	32332	11128	94.73	1.98	11	23597	9648	95.00	1.30	26
	11	27689	16262	95.48	1.36	7	25696	9948	92.77	1.65	27
	16	**36500**	**9404**	94.33	1.84	13	**19316**	**6216**	94.22	1.74	29
	21	48349	12648	95.11	1.62	7	24797	8648	93.83	0.94	28
	26	51984	15648	94.44	1.70	9	30078	7223	94.47	1.38	24
CoSyNE	1	**30924**	**8578**	95.26	0.65	10	**54234**	**12248**	92.58	0.97	29
	6	73780	12648	92.98	2.14	9	63119	14648	90.00	2.07	18
	11	68040	16485	92.76	1.11	4	74635	19559	88.00	1.63	5
	16	36720	0	94.73	0	1	65178	0	92.50	0	1
	21	59400	0	94.73	0	1	–		–		0
	26	77760	4860	92.10	3.64	2	–		–		0

3.3 Analysis and Discussion

In Table 2, for the Iris problem, the NSP shows to outperform the other methods in terms of least number of function evaluations and best success rate. The CoSyNE takes the most training time and gives a poor success rate. The CME performs slightly better than the CoSyNE but tends to be weaker than the NSP. Therefore, the NSP is the best choice for this problem. The depth of search of NSP and CME does not make a significant difference. In the case of CoSyNE,

Table 3. Performance in Heart and Breast Cancer classification problems

		Heart					**Breast Cancer**				
Method	Depth	FuncEval		Test Error		Success	FuncEval		Test Error		Success
NSP	1	**36740**	**2093**	78.10	1.04	28	12158	3770	97.35	0.35	30
	6	39516	1779	76.89	1.03	28	11113	4128	97.05	0.45	30
	11	38489	4629	78.00	0.99	28	11044	4785	97.24	0.50	30
	16	**38267**	**8236**	78.75	1.03	29	**10485**	**3071**	97.16	0.29	30
	21	40316	9648	77.82	0.82	28	12830	8111	97.20	0.45	30
	26	42768	9544	77.78	1.12	28	11469	3666	97.23	0.47	30
CME	1	**60240**	**9271**	79.14	1.15	14	**39221**	**2950**	97.83	0.45	16
	6	59252	9338	78.84	0.82	13	42221	3250	97.41	0.38	13
	11	58917	9117	78.90	0.92	20	38221	2885	97.33	0.44	14
	16	53718	8848	80.29	0.85	17	41229	2879	97.63	0.38	15
	21	56867	8791	79.34	0.81	12	38771	2734	97.67	0.51	17
	26	58713	9867	79.29	0.80	14	44244	3150	97.81	0.45	12
CoSyNE	1	**80320**	**9271**	77.14	1.55	2	–		–		0
	6	–		–		0	–		–		0
	11	–		–		0	–		–		0
	21	–		–		0	–		–		0
	26	–		–		0	–		–		0

the depth of search plays a significant role. The large depths fail to deliver good solutions in terms of training time and success rate.

In the results for the Wine classification problem in Table 2, the depth of search for NSP and CME does not play a significant role; however, it shows significance in CoSyNE, which show optimal results in depth of 1 generation only. The best results are given by NSP in terms of least training time and best success rate.

The results from the Heart classification problem in Table 3 show that the NSP method outperforms CME and CoSyNE. The CoSyNE method delivered a solution only with the depth of 1 generation only. It performed poorly when compared to CME and NSP.

The depth of search in NSP and CME does not show significant difference. Similar performance is shown for the Cancer problem, where NSP outperforms other methods. The depth of search for NSP is not significant here. In general, for all the datasets, NSP achieves the highest success rate in comparison to CoSyNE and CME. CoSyNE showed the weakest performance in terms of success rate and function evaluations.

The results in general show that that a lower depth of search used for a subcomponent is efficient especially for CoSyNE. The depth of search in CME and NSP does not show a significant difference for the interval of [1, 26]. This is due to the difference in the encoding schemes for the respective CC frameworks which use different number of subcomponents in the three paradigms. The difference in the number of subcomponents used in CME and NSP is small. Therefore, the difference in the depth of search in both methods are not so big.

In summary, the number of subcomponents used in the CC framework directly influences the training problem. The NSP and CME perform well regardless of the depth of search, while the CoSyNE shows optimal performance for the depth of 1 generation only. The CoSyNE uses a relatively higher number of subcomponents when compared to CME and NSP.

CoSyNE showed impressive results for pole balancing problems in [9], however, its contribution in pattern recognition problems is weak. This is due to the nature of the problem as the pole balancing problem is a control problem.

The pole balancing problem is a dynamical system which requires a higher level of dependency among the weights of the neural network. Therefore, each synapse in the neural network has to be adjusted on its own without affecting the other synapses. CoSyNE showed to be the best algorithm for this problem as it has the feature of independently adjusting each synapse which was natural to their encoding scheme. In CoSyNE, each subcomponent has the greatest flexibility of adaptation without affecting other subcomponents in comparison to NSP and CME. Therefore, it is better for highly dependent problems such as the pole balancing problem. The pattern recognition problems studied in this paper do not have the same level of dependency, and therefore, CoSyNE has not shown good results.

Note that NSP has also delivered similar generalisation performance in comparison to CME and CoSyNE while achieving a better training performance. In Tables 2 and 3, the generalisation performance is not significantly different for CME and NSP for all problems.

4 Conclusions and Future Work

This work introduced a new encoding scheme (NSP) for building subcomponents in the cooperative coevolution framework. The paradigm was tested on four benchmark problems and compared to two different encoding schemes used in general (CME and CoSyNE) from literature.

An important question raised from this research was to evaluate the optimal depth of search in the subcomponents for the respective paradigms. The results show that the depth of search is sensitive for CoSyNE only. The depth of search for 1 generation only gives acceptable results in the CoSyNE algorithm in all the given problems. However, these results were poor when compared to CME and NSP. NSP and CME performs relatively well in the given depths (1 - 26 generations).

In general, the NSP has shown to outperform the other methods in terms of the number of function evaluations and the success rate for all the given problems. Even if the NSP is combined with the cost of evaluation in the initialisation stage, it performs better than the other methods and at the same time, it provides a similar generalisation performance to CME and CoSyNE.

Future work can examine the implementation of the NSP in evolving the weights and the network architecture at the same time during training. A paradigm where the different encoding schemes can be combined during training

can also be explored. This would be implemented by increasing the modularity by increasing the number of subcomponents as the algorithm is progressing towards the final solution.

References

1. Potter, M.A., Jong, K.A.D.: A cooperative coevolutionary approach to function optimization. In: PPSN III: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature, London, UK, pp. 249–257. Springer, Heidelberg (1994)
2. Yang, Z., Tang, K., Yao, X.: Large scale evolutionary optimization using cooperative coevolution. Inf. Sci. 178(15), 2985–2999 (2008)
3. Potter, M.A., De Jong, K.A.: Cooperative coevolution: An architecture for evolving coadapted subcomponents. Evol. Comput. 8(1), 1–29 (2000)
4. García-Pedrajas, N., Ortiz-Boyer, D.: A cooperative constructive method for neural networks for pattern recognition. Pattern Recogn. 40(1), 80–98 (2007)
5. Garcia-Pedrajas, N., Hervas-Martinez, C., Munoz-Perez, J.: COVNET: a cooperative coevolutionary model for evolving artificial neural networks. IEEE Transactions on Neural Networks 14(3), 575–596 (2003)
6. Garcia-Pedrajas, N., Hervas-Martinez, C., Munoz-Perez, J.: Multi-objective cooperative coevolution of artificial neural networks (multi-objective cooperative networks). Neural Netw. 15(10), 1259–1278 (2002)
7. Gomez, F., Mikkulainen, R.: Incremental evolution of complex general behavior. Adapt. Behav. 5(3-4), 317–342 (1997)
8. Gomez, F.J.: Robust non-linear control through neuroevolution. Technical Report AI-TR-03-303, PhD Thesis, Department of Computer Science, The University of Texas at Austin (2003)
9. Gomez, F., Schmidhuber, J., Miikkulainen, R.: Accelerated neural evolution through cooperatively coevolved synapses. J. Mach. Learn. Res. 9, 937–965 (2008)
10. Asuncion, A., Newman, D.: UCI Machine Learning Repository (2007)
11. Deb, K., Anand, A., Joshi, D.: A computationally efficient evolutionary algorithm for real-parameter optimization. Evol. Comput. 10(4), 371–395 (2002)
12. CantuPaz, E., Kamath, C.: An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems. IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics 35(5), 915–933 (2005)

Evolving Cooperation in the Spatial N-player Snowdrift Game

Raymond Chiong and Michael Kirley

Department of Computer Science and Software Engineering,
The University of Melbourne, Victoria 3010, Australia
{rchiong,mkirley}@csse.unimelb.edu.au

Abstract. The Snowdrift game is a well-known social dilemma model frequently used in evolutionary game theory to investigate the emergence of cooperative behaviour under different biologically or socially plausible conditions. In this paper, we examine a multi-player version of the Snowdrift game where (i) the agents playing the game are mapped to the nodes of a regular two-dimensional lattice, (ii) the number of rounds of the game varies from a "one-shot" version to a fixed number of repeated interactions, and (iii) a genetic algorithm is used to evolve agent actions (strategy update) over a fixed number of generations. Comprehensive Monte Carlo simulation experiments show that cooperative behaviour is promoted in the multi-player iterated Snowdrift game. This emergent behaviour may be attributed to the combination of spatial reciprocity, based on the inherent capabilities of the genetic algorithm to explore the diverse sets of agents' strategies, and repeated interactions. The simulation results also uncover some interesting findings regarding the effect of repeated interactions in the game.

1 Introduction

Understanding the emergence and maintenance of cooperation among competing entities has been a central theme in many scientific disciplines, including biology, physics, computer science, operations research, economics, social and political sciences. Boosted by the seminal books of Maynard Smith [11] and Axelrod [1], evolutionary game theory has been used extensively as a standard tool for studying cooperative phenomena. Social dilemma games, such as the Prisoner's Dilemma (PD) and the Snowdrift (SD) game, constitute powerful metaphors describing situations characterised by varying degrees of conflicting interests between individuals. In both games, players have an opportunity to either *cooperate* or *defect*, with cooperators making a sacrifice to benefit others at some cost, yet defectors reap the benefit without needing to bear any cost. However, if every player defects, the outcome becomes worse than if they had cooperated.

While the PD game reflects such a social dilemma in the strictest sense, the SD game relaxes some constraints by (i) allowing players to obtain some immediate benefit from their cooperative acts and (ii) sharing the cost of cooperation between cooperators.

J. Li (Ed.): AI 2010, LNAI 6464, pp. 263–272, 2010.

Table 1. The payoff matrix for the two-player SD game

	Cooperate	Defect
Cooperate	$b - c/2$	$b - c$
Defect	b	0

A simple narrative can be used to illustrate the basic SD game: two drivers are on their way driving home, but get trapped on either side of a snowdrift. Both of them have the options of staying in the car and do nothing (i.e. defect) or coming off and start shovelling the snowdrift (i.e. cooperate). If they cooperate, both can enjoy the benefit of getting home while sharing the work. Opting to defect and let the other one do all the work is the best option, but if both think the same then no one gets home. Table 1 shows the payoff matrix for the two-player SD game, where b is the benefit and c is the cost of cooperation. Here, $b > c > 0$.

The seemingly minor difference between the PD and the SD game has led to some significant changes in the behavioural outcome (see [5] for a review). In particular, spatial structure [12,13] that is widely known to be beneficial for the evolution of cooperation in the PD literature may be detrimental to cooperation in the SD game. In their attempt to impose spatial structure on the SD game, Hauert and Doebeli [7] observed that spatial extension generally fails to promote cooperative behaviour. They showed that the fraction of cooperators becomes less for a wide range of parameters within the spatial settings, as contrary to the non-spatial SD game where intermediate levels of cooperation typically persist. In particular, cooperation could be fully eliminated if the cost-to-benefit ratio of cooperation is high.

While intriguing, several studies thereafter (e.g. [15,16,17]) have identified a number of important factors that influence the results of [7], one of which is the role of update rules (i.e., how players' strategies are being updated). In one of Nowak and May's earlier works on spatial models (see [13]), they used a simple imitation-based strategy update mechanism and found that cooperation can be supported in the Hawk-Dove game – a game with the same payoff ranking but a slightly different matrix structure to the SD game. Hauert and Doebeli, on the other hand, employed the replicator rule. Sysi-Aho et al. [15] studied the effect of changing the strategy update rules in the spatial SD game similar to that discussed in [7], and found that their evolution-based rules result in cooperation levels which differ largely from those obtained using the replicator dynamics. In the model developed by Sysi-Aho et al., the rules have been defined in such a way that changes in players' strategies are determined by each player's decisions (considering the strategies of other players) within the local neighbourhood. With these strategy evolution rules, cooperation persists through the whole temptation parameter range.

Motivated by these findings, this paper proposes the use of Genetic Algorithms (GAs) [8] as the strategy update rule to play the spatial N-player SD game. Using GAs to evolve cooperation in PD games is not uncommon in the literature, most notable being the original work from Axelrod himself (see [2]). To the best of our

knowledge, however, we are not aware of any work that has used GAs for the SD game, especially the N-player version. It would therefore be interesting to see how the behavioural outcome of the game is changed when GAs are adopted. In addition, we also propose a new extension to the N-player SD game, which we called the N-player Iterated SD (N-ISD) game. The idea of this N-ISD game is similar to that of the Iterated PD (IPD), where a single game consists of a number of rounds of the simple SD game instance.

The remainder of this paper is organised as follows: Section 2 briefly introduces background information related to the N-player SD game. In Section 3, we present the details of our model. Section 4 describes the experimental settings and results. Finally, we draw conclusions in Section 5 and highlight some potential future work.

2 The N-player Snowdrift Game

The N-player SD game is a generalisation of the two-player SD game, in which a group of players ($N > 2$) simultaneously interact with one another rather than only two playing against each other. Following the storyline previously given, we can imagine a scenario where several drivers are now trapped at a snowdrift occurring in the middle of a cross-road. Similarly, everybody wants to go home, but most likely not all would be willing to do the laborious shovelling job. The ideal situation is for every driver to shovel, as all will share the labouring cost and get home in the quickest manner. However, there are always free-riders who would not want to do anything but take advantage of the efforts of others. The worst case being no one carries out the job and all get stuck.

Despite a large volume of publications focused on the two-player SD game from different fields, the N-player version has only recently been studied in detail (e.g. [3,4,9,10,14,19]). Typically, these studies have examined the non-spatial game with well-mixed populations. Two notable exceptions, however, have considered spatially structured populations (see [4] and [10]). The fundamental difference between the N-player game and the two-player one is that the former concerns multi-player interactions while the latter on pair-wise interactions. As pointed out by Gokhale and Traulsen [6], although many results from pair-wise interactions can be generalised to multiple players, as far as the *two players and two strategies* (2×2) game is concerned, statements derived for pair-wise interactions do not hold for N-player games with more than two strategies.

3 The Model

We consider the spatial N-player SD game, where interacting individuals (or agents) are placed on the vertices of a 30×30 square lattice with periodic boundary conditions. The population of agents are initialised with random strategies and are then randomly distributed on the sites of the lattice. Each agent plays an iterative game with its local neighbours at every generation. The utility (fitness) of each agent is determined by summing its payoffs in the game against

the neighbours. At the end of each generation, all agents are presented with an opportunity to update their strategies according to the payoffs received. The payoffs are calculated according to the following utility function, U:

$$U = \begin{cases} b \times i - c \times (N-1)/i & \text{for cooperators,} \\ b \times i & \text{for defectors.} \end{cases} \quad (1)$$

where b is the benefit, c is the cost of cooperation, and i is the number of cooperators.

3.1 Strategy Representation

The most straightforward strategy representation for the game is the scheme proposed by Axelrod for the IPD game [2]. However, as pointed out by Yao and Darwen [18], Axelrod's representation scheme does not scale well with the increase in the number of players for N-player games. In addition, it also includes redundant information by identifying which of the other players cooperated or defected, whereas the only information requires is the actual number of cooperators or defectors. As such, we have decided to adopt the representation developed by Yao and Darwen instead.

Under this representation, a history of l rounds for an agent can be represented as the combination of the following bit strings:

- l bits to represent the agent's l previous actions, where '1' = defection and '0' = cooperation.
- $l \times log_2 N$ bits to represent the number of cooperators in the previous l rounds among the agent's social group, where N is the group size.

Based on preliminary empirical analysis, we have limited the number of previous actions in memory to 3 (i.e. $l = 3$), as this value can be used to generate a very large set of possible strategies. In the case of $N = 4$, for example, the history for an agent would be $3 + 3 \times log_2 4 = 9$ bits long based on this representation scheme.

Figure 1 illustrates a possible history an agent could have. The initial three bits are the agent's previous three actions. From the figure we can see that the agent defected in the last two rounds and cooperated the round before that. The two-bit sets after the first three bits represent the number of cooperators in the

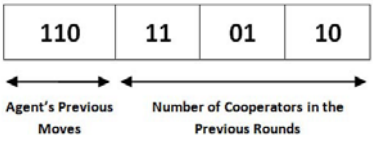


Fig. 1. History of an agent: the first three bits are the agent's previous moves; the subsequent bit sets represent the number of cooperators in the previous rounds

last three rounds from the agent's social group. This agent's history indicates that there were 3, 1 and 2 cooperators in the agent's group in the last three rounds.

An agent's strategy is used to identify an action (or move) at the next interaction, in response to the history. Therefore, when $N = 4$ the strategy should be at least $2^9 = 512$ bits in length. Using the example from Figure 1, the history 110 11 01 10 would trigger the agent to make a move corresponding to the bit listed in the 438^{th} position of its strategy string (438 is the decimal number for the binary 110110110).

We have added an additional three bits to each strategy to compensate for the lack of complete history in the first three rounds. This means that the actions in the first three rounds of each generation are hard-coded into the strategy. Thereafter, the moves are made based on the history of the agent and its group members. It is important to note that as the group size increases more bits are needed for the strategy representation.

3.2 Strategy Update

Strategy update is an evolutionary mechanism in this study. Here, we use a GA to evolve the pool of agents' strategies. Each agent plays the game repeatedly for T iterations at each generation. Every agent uses a unique strategy to decide the action to play at iteration t, where $t \in [1..T]$. At the end of T iterations, agents may change their behaviour by comparing their utility to that of neighbouring agents and employing standard evolutionary operators. For crossover, a random number is generated to determine whether it should take place. Two-point crossover with rank-based selection is used, where 60% of the best strategy within a group is being selected and recombined with 40% of the current strategy of an agent. Note that this will happen only when the crossover rate is satisfied and the current strategy is ranked below the elite group (in this study, strategies that rank among the top 50% are considered to be in the elite group). Otherwise, nothing comes about. This elite preserving mechanism ensures that good strategies are being carried forward to the next generation. Finally, a random number is generated to determine whether a strategy will be mutated. A mutation consists of random bit-flip in the strategy's bit representation.

4 Experiments and Results

The underlying hypothesis tested in this study was that the use of GAs would promote high levels of cooperation in the spatial N-player SD game. Using Monte Carlo simulations, we have investigated the behavioural outcome of several parameter settings. Three separate sets of simulation experiments have been conducted. The settings of each set of experiments and their corresponding results are discussed in this section.

The first set of experiments concerned two variants of our model, one based on the spatial N-player SD game while the other is the extended spatial N-ISD game.

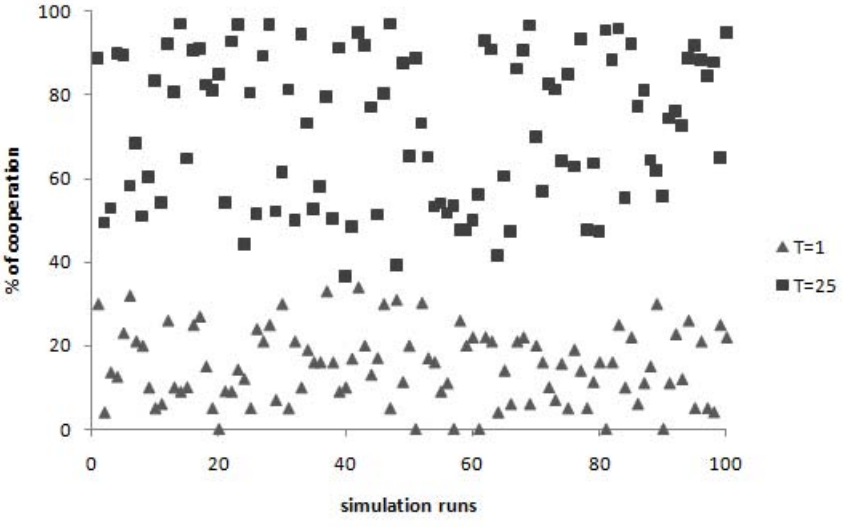


Fig. 2. The levels of cooperation with $N = 5$ over 100 simulation runs for the spatial N-player SD ($T = 1$) game and the spatial N-ISD ($T = 25$) game

The aim here was twofold: to examine the ability of GAs for evolving cooperative behaviour in general, and to see how the introduction of repeated interactions within each generation changes the behavioural outcome. The specific model parameters are as follows:

- The payoff values were calculated using Eq. 1 with $b = 3$ and $c = 1$
- All the agents played against one another for 500 generations, with $T = 1$ for the spatial N-player SD game and $T = 25$ for the spatial N-ISD game
- The crossover rate was set to 0.7 and the mutation rate to 0.05 for the GA
- The interaction topology was based on the well-known *von Neumann* neighbourhood structure
- The results reported, were average over 100 simulation runs

In order to clearly identify the impact that spatial extension has in our model, we have also performed additional simulation runs using the same settings described above (except for the interaction topology) on non-spatial versions of the N-player SD game and the N-ISD game. In these non-spatial games, the notion of local neighbourhood and the neighbours does not exist, consequently the other players (N was still fixed to 5) were randomly selected from anywhere in the population.

Figures 2 and 3 show the results of this series of experiments. As can be seen from Figure 2, cooperation is the dominant strategy when the players are engaged in the spatial N-ISD game. In the majority of the runs, we have more than 50% of cooperators in the population. In fact, many runs ended up with more than 80% of cooperation. There were only a negligible few instances where the cooperation level is around 40%.

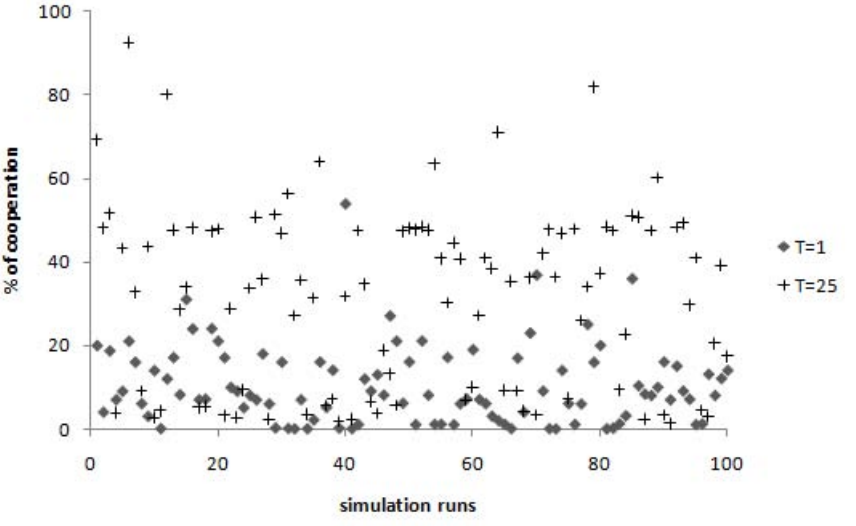


Fig. 3. The levels of cooperation with $N = 5$ over 100 simulation runs for the non-spatial N-player SD ($T = 1$) game and the non-spatial N-ISD ($T = 25$) game

In contrast, the levels of cooperation achieved are quite poor in the spatial N-player SD game. Most of the runs ended up with less than 20% of cooperators in the population. Particularly, there were a number of runs where cooperators were completely wiped out. The results indicate that GAs require more learning interactions within each generation to discover cooperative strategies.

Figure 3 displays the results of non-spatial N-player SD and N-ISD games. We see that, in the non-spatial N-ISD game, a substantial amount of runs ended up with 50% of cooperators or less. This is consistent with the literature in evolutionary game theory predicting that there exists an intermediate level of cooperation in the typical two-player SD game. In the non-spatial N-player SD game, however, many runs finished with 0% of cooperators.

From these results, it is clear that spatial models are more likely to promote higher levels of cooperation than non-spatial ones for the SD game with N-player interactions when GAs are used as the strategy update rule. Repeated interactions in each game appear to be an essential characteristic fostering highly cooperative behaviour. Based on these results, the next question to be asked is: does this mean that an increase in the number of interactions in each generation would lead to even higher levels of cooperation? This brings us to the second set of experiments conducted, aiming to find out more about the effect of repeated interactions in the spatial N-ISD game. To do so, we run the experiments again using the same settings as in the first set, but this time we have varied the values of T from 25 to 50, 75 and 100.

Figure 4 shows the levels of cooperation achieved when the number of repeated interactions within each generation was varied. Surprisingly, the result indicates that when the value of T increases, the ratios of cooperation decrease. This

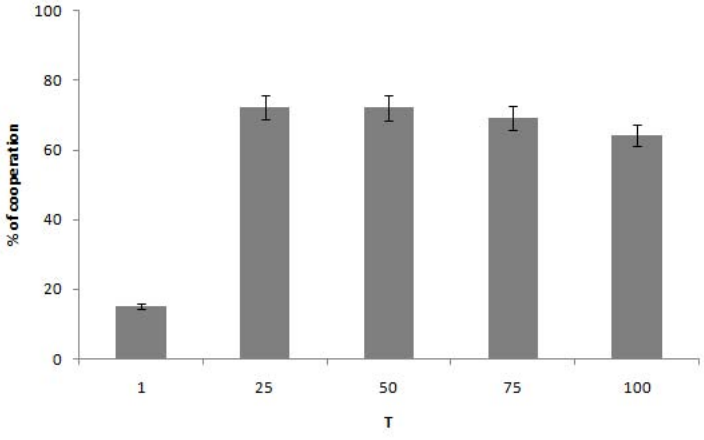


Fig. 4. The levels of cooperation with $N = 5$ and varying values of T averaged over 100 simulation runs

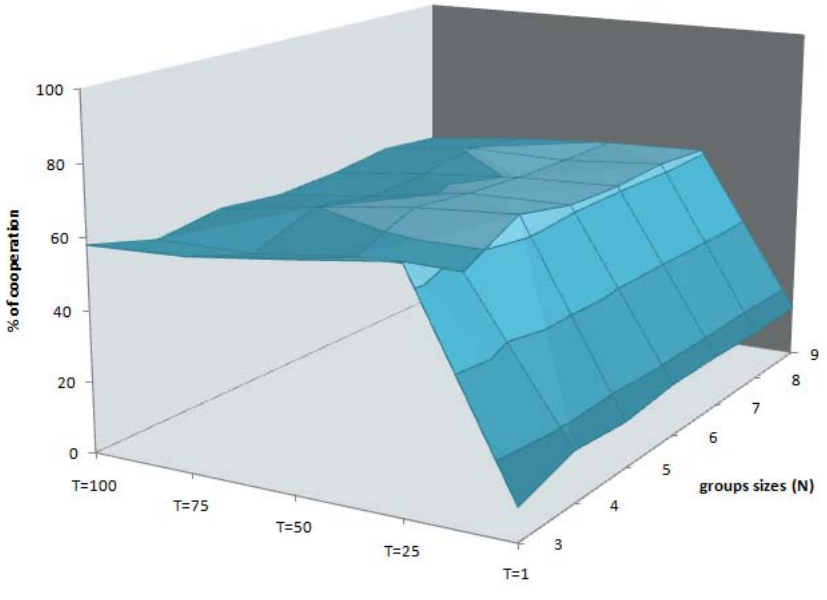


Fig. 5. The levels of cooperation with varying values of T and group sizes averaged over 100 simulation runs

result is in contrast to that expected based on the notion of *direct reciprocity* that has served the IPD game so well. To verify whether the differences in results were significant or not, we have performed t-tests on the 100 individual runs to compare $T = 25$ against $T = 50$, 75 and 100. The p values confirm that, while the results of $T = 25$ and $T = 50$ are insignificant, $T = 25$ vs. $T = 100$ are statistically significant ($p = 0.0143$).

In the next stage of simulation experiments, we have repeated the previous experiments but this time vary the group sizes from $N = 3$ to 9. The purpose was to see if the effects of T observed above (see Figure 4) were consistent across different group sizes. All parameter settings remained identical, except for the way in which neighbours were selected. Instead of a fixed neighbourhood structure, this time, the neighbours were randomly selected from the immediate neighbourhood at the beginning of each run (but did not change thereafter).

Figure 5 shows the simulation results of different T values across different group sizes averaged over 100 simulation runs. Generally, we see that the trends are consistent when the values of N increase from 3 to 9. In all cases, we observe that the number of cooperators dropped regularly when agents were engaged in more number of rounds per game. In terms of the levels of cooperation, most of the settings yield more than 50% of cooperators when repeated interactions are enforced. On a side note, we notice that the cooperation rates of $N = 5$ with fixed neighbourhood across T values (see Figure 4) are higher than other group sizes with randomly picked neighbours.

5 Conclusion and Future Work

In this paper, we have investigated the use of GAs to evolve cooperative behaviour in the spatial N-player SD game. Detailed computational simulations have shown that repeated interactions are essential for GAs to learn and evolve agents' strategies to a more cooperative state. A somewhat counter intuitive result observed in the simulation trials was that as the agents learn more about the game (as a direct result of increasing the number of iterations), there was a corresponding drop in the levels of cooperation. Obviously, achieving high levels of cooperation is challenging in multi-player games, especially for larger values of N. However, we suggest that the trend in the results can be best explained based on the payoff structure of the SD game – cooperation being the better option when the opponent defects. For large values of T, some agents learn that they could actually free-ride on others' efforts, hence turning away from cooperating.

Future work will further investigate the effects of repeated interactions across different evolutionary games.

References

1. Axelrod, R.: The Evolution of Cooperation. Basic Books, New York (1984)
2. Axelrod, R.: The Evolution of Strategies in the Iterated Prisoner's Dilemma. In: Davis, L. (ed.) Genetic Algorithms and Simulated Annealing, pp. 32–41. Morgan Kaufmann, Los Altos (1987)
3. Chan, C.H., Yin, H., Hui, P.M., Zheng, D.F.: Evolution of Cooperation in well-mixed N-person Snowdrift Games. Physica A: Statistical Mechanics and its Applications 387, 2919–2925 (2008)
4. Chen, X.-J., Wang, L.: Effects of Cost Threshold and Noise in Spatial Snowdrift Games with Fixed Multi-person Interactions. Europhysics Letters 90, 38003 (2010)

5. Doebeli, M., Hauert, C.: Models of Cooperation based on the Prisoner's Dilemma and the Snowdrift Game. Ecology Letters 8, 748–766 (2005)
6. Gokhale, C.S., Traulsen, A.: Evolutionary Games in the Multiverse. Proceedings of the National Academy of Sciences of USA 107, 5500–5504 (2010)
7. Hauert, C., Doebeli, M.: Spatial Structure often Inhibits the Evolution of Cooperation in the Snowdrift Game. Nature 428, 643–646 (2004)
8. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. MIT Press, Cambridge (1992)
9. Ji, M., Xu, C., Zheng, D.F., Hui, P.M.: Enhanced Cooperation and Harmonious Population in an Evolutionary N-person Snowdrift Game. Physica A: Statistical Mechanics and its Applications 389, 1071–1076 (2010)
10. Lee, K.H., Chan, C.H., Hui, P.M., Zheng, D.F.: Cooperation in N-person Evolutionary Snowdrift Game in Scale-free Barabási–Albert Networks. Physica A: Statistical Mechanics and its Applications 387, 5602–5608 (2008)
11. Maynard Smith, J.: Evolution and the Theory of Games. Cambridge University Press, Cambridge (1982)
12. Nowak, M.A., May, R.M.: Evolutionary Games and Spatial Chaos. Nature 359, 826–829 (1992)
13. Nowak, M.A., May, R.M.: The Spatial Dilemmas of Evolution. International Journal of Bifurcation and Chaos 3, 35–78 (1993)
14. Souza, M.O., Pacheco, J.M., Santos, F.C.: Evolution of Cooperation under N-person Snowdrift Games. Journal of Theoretical Biology 260, 581–588 (2009)
15. Sysi-Aho, M., Saramäki, J., Kertész, J., Kaski, K.: Spatial Snowdrift Game with Myopic Agents. The European Physical Journal B 44, 129–135 (2005)
16. Számadó, S., Szalai, F., Scheuring, I.: The Effect of Dispersal and Neighbourhood in Games of Cooperation. Journal of Theoretical Biology 253, 221–227 (2008)
17. Wang, W.-X., Ren, J., Chen, G., Wang, B.-H.: Memory-based Snowdrift Game on Networks. Physical Review E 74, 056113 (2006)
18. Yao, X., Darwen, P.: An Experimental Study of N-person Iterated Prisoner's Dilemma Games. Informatica 18, 435–450 (1994)
19. Zheng, D.F., Yin, H.P., Chan, C.H., Hui, P.M.: Cooperative Behavior in a Model of Evolutionary Snowdrift Games with N-person Interactions. Europhysics Letters 80, 18002 (2007)

Sampling Methods in Genetic Programming for Classification with Unbalanced Data

Rachel Hunt[1], Mark Johnston[1], Will Browne[2], and Mengjie Zhang[2]

[1] School of Mathematics, Statistics and Operations Research
[2] School of Engineering and Computer Science
Victoria University of Wellington, P.O. Box 600, Wellington, New Zealand

Abstract. This work investigates the use of sampling methods in Genetic Programming (GP) to improve the classification accuracy in binary classification problems in which the datasets have a class imbalance. Class imbalance occurs when there are more data instances in one class than the other. As a consequence of this imbalance, when overall classification rate is used as the fitness function, as in standard GP approaches, the result is often biased towards the majority class, at the expense of poor minority class accuracy. We establish that the variation in training performance introduced by sampling examples from the training set is no worse than the variation between GP runs already accepted. Results also show that the use of sampling methods during training can improve minority class classification accuracy and the robustness of classifiers evolved, giving performance on the test set better than that of those classifiers which made up the training set Pareto front.

1 Introduction

Classification is the act of determining which class an instance of a dataset belongs to, based on its properties. Class imbalance occurs when there are a large number of instances in one class (the majority class) and only a small number of instances in the other (the minority class). Imbalance is common in many situations, including medical diagnosis [5] and fraud detection [8]. In these and many class imbalance situations, it is often the minority class that is of interest, and has a higher misclassification cost than the majority class.

In standard Genetic Programming (GP) the overall classification accuracy is often used as a measure of a program's fitness (ability to correctly classify instances) during the evolutionary process. Standard GP trains the population of programs on every training dataset instance at each generation. The fitness function is the classification accuracy of the program. In situations with class imbalance this standard approach often causes a bias towards correctly classifying the majority class. A program can have a high fitness, yet rarely classify a minority class instance correctly.

Methods of approaching the class imbalance problem in genetic programming fall into two main categories: solutions at the data level; and solutions at algorithm level (including the multi-objective approach). Solutions at the data level

J. Li (Ed.): AI 2010, LNAI 6464, pp. 273–282, 2010.

involve re-sampling the data to artificially balance the dataset, often through various forms of under and over sampling. Under-sampling uses fewer than the total number of majority class instances and over-sampling replicates minority class examples. If outlying data instances are repeated they can skew the distributions. Under-sampling can lead to the loss of potentially useful information.

The goal of this work is to address the class-imbalance problem by developing new sampling methods that lead to the evolution of more robust classifiers that improve the classification performance of genetic programming on unbalanced datasets and to increase the successful classification rate of the minority class to be similar to the majority class classification accuracy and as high as the dataset allows, as lack of data, missing values and noise may also affect accuracy as well as class imbalance.

The rest of this paper is organised as follows. Section 2 describes the background work relating to this paper. Section 3 describes the new sampling methods used for classification tasks with unbalanced datasets. Section 4 describes the experimental design and Section 5 discusses the results. Section 6 concludes the paper.

2 Background

Gathercole and Ross [4] use Dynamic Subset Selection (DSS), Historical Subset Selection (HSS) and Random Subset Selection (RSS) on a large unbalanced dataset. DSS randomly selects a sample from the training set of size N with a bias towards those instances which are often misclassified, or have not been included in the subset for several generations. HSS uses standard GP runs on the training dataset to determine how 'difficult' each instance is by the number of times it is mis-classified by the best population member in each run. The cases with greater 'difficulty' are then used as the sample subset for HSS runs. RSS selects a random subset of data instances at each generation, each data instance has equal probability of being selected. In each method, the subset size fluctuates around the target subset size.

Doucette and Heywood [3] use a simplified version of DSS, which they call the Simple Active Learning Heuristic (SALH). This is used alone and then in combination with a fitness function based on the Wilcoxon-Mann-Whitney (WMW) statistic which approximates the *area under the receiver operating curve* (AUC). Subsets were generated by selecting from the training set, with uniform probability, an equal number of instances from both minority and majority classes. The combined use of the WMW fitness function and sampling method was shown to greatly increase the AUC. The AUC is a robust measure of how well the classifier can discriminate between the two classes and is approximated by the WMW [9].

Iba [6] applies the re-sampling techniques of boosting and bagging to GP. In both techniques the entire population of instances is split into subpopulations. The best individuals from each vote to create a composite tree output. These techniques use re-sampling to get different training sets for each classifier. In bagging, the training set is the size of all the training instances, but instances may appear multiple times or not at all. In boosting, every instance is weighted

Table 1. Number of samples and sample size for each dynamic sampling method

Method	Number of Samples	Sample Size
Basic Under	1	$2N_{min}$
Under A, Under B	$\lfloor \frac{N_{maj}}{N_{min}} + \frac{1}{2} \rfloor$	$2N_{min}$
Basic Over	1	$2N_{maj}$
Over A, Over B	$\max\{2, \frac{1}{2}\lfloor \frac{N_{maj}}{N_{min}} + 0.5 \rfloor\}$	$2\lfloor \frac{N_{maj}}{NumSamples} \rfloor$

to reflect its importance and classifiers have differing voting strengths based on their accuracy. Paris et al. [7] also apply boosting to GP with success in both regression and binary classification problems.

3 New Sampling Methods

Static Sampling Before Training. We wish to establish that *during training* the variation in performance introduced by random sampling of training examples is no worse than the variation already inherent due to random selection of the initial population of trees and random selections made during GP runs. To achieve this, at the beginning of the evolution process we sampled with uniform probability (alternately from each class) from the training dataset until we had a balanced sample (equal numbers of majority and minority class instances) of the desired size. This sample was taken without replacement and then used in training to evaluate each program in the population of programs at every generation.

Dynamic Sampling During Training. We used six sampling methods to create balanced samples from the training dataset. At each generation a new sample is taken and used to evaluate the current population of genetic programs. The number of samples and size of each sample is given in Table 1.

Basic Under-sampling. Data instances from the training dataset are sampled without replacement to make up a balanced sample.

Each minority class instance is selected, and then an equal number of instances from the majority class are randomly selected for the sample.

Basic Over-sampling. All the majority class instances from the training dataset are used, and minority class instances are sampled repeatedly until there are equal numbers of majority and minority class instances. Each data instance is in the sample.

Under-sampling A. Multiple balanced samples are created at each generation, each using all the minority class instances of the training dataset. Each sample contains all minority class and an equal number from the majority class, the majority class is sampled without replacement. Each training set data instance is used in each generation. Each program in the population is run on all of the sample sets, and the fitness function is the average fitness of the program across all the sample sets.

Under-sampling B. Uses the same sampling method as Under-sampling A. The fitness function is taken to be the minimum 'fitness' of the program across all the sample sets.

Over-sampling A. Over-samples the minority class and under-samples the majority class. Multiple balanced samples are created at each generation. At each generation, each program in the population is run on all of the sample sets, and the fitness function is the average fitness of the program across all the sample sets. In each generation of evolution each training dataset instance is used.

Over-sampling B. Uses the same sampling method as Over-sampling A. The fitness function is taken to be the minimum fitness of the program across all the sample sets.

Basic Under and Over sampling use a balanced subset of fixed size, in which instances are selected with uniform probability without replacement. This is essentially the SALH used by Doucette and Heywood [3], although it is unclear if SALH samples with or without replacement. This is similar to RSS [4] in that the sample is taken at each generation and instances have equal probability of being selected within each class.

Over-sampling and Under-sampling methods A and B take multiple SALH samples and evaluates the population of genetic programs at that generation on all samples. The fitness of each program is then assigned as either the minimum fitness recorded across the samples (B), or as the average fitness recorded across all samples (A). In either case, the fitness is to be maximized.

4 Experimental Design

Datasets. The following three datasets were chosen from the *UCI machine learning repository* [1].

Yeast. This dataset has 1484 instances, each with eight attributes. This data is used for classifying the localization site of proteins in yeast cells. There are 10 possible sites, of which we selected three different sites (MIT, ME1 and POX) to act as the minority class. MIT, ME1 and POX have 244, 44 and 20 instances respectively, giving imbalance ratios of 1:5, 1:33 and 1:73.

SPECT Heart. This dataset represents the classification of diagnosis of cardiac Single Proton Emission Computed Tomography (SPECT) images. This is a binary classification problem; a patient was classed as either normal or abnormal. The dataset contains 267 SPECT image sets (patients), of which 55 represent abnormal diagnosis (20.6%). The dataset was processed to obtain 22 binary features. This is an approximate class imbalance of 1:4.

Pima Indians Diabetes Database. This dataset has 768 data instances, each representing a female of Pima Indian heritage over the age of 21. This is a binary classification task whether each patient shows signs of diabetes, based on the World Health Organisation criteria. There are eight attributes, information collected from each patient. There are 768 instances with 268 showing signs positive for diabetes (34.9%). This gives an approximate class imbalance of 1:2.

Functions and Terminals. The terminal set represents the input into the GP system. For each dataset their attributes and constants were used as terminals. Constant terminals were random numbers generated at the start of the evolutionary process. The function set is $\{+, -, \times, \%, \texttt{if}\}$. The arithmetic operators take two arguments. The first three arithmetic operators, $+$, $-$, $\times$, have their usual meanings. The % is as usual division except when dividing by zero where the value returned is zero. The `if` function takes three arguments, if the first argument is negative then it returns the second, else the third is returned.

Evolutionary Parameters. A population of 500 programs was used, the initial population was generated using the ramped half-and-half method and then evolved for 50 generations, unless a solution was found early in which case evolution was terminated. The genetic operators used were crossover, mutation and reproduction (elitism), at rates of 60%, 30% and 10% respectively. The selection operator was tournament selection, in which a tournament of size seven was used to select individuals for the genetic operators. The maximum depth of each program was six in order to restrict bloat effects.

5 Experimental Results

Results for Standard GP. The results of standard GP on the datasets are presented in Table 2. It is apparent that results are heavily biased towards the majority class. The classification accuracy on the minority class of the SPECT dataset is extremely low, only 4.1%, which shows that the standard fitness function is not appropriate for this case. From this it is clear that overall accuracy is unlikely to be a good measure of accuracy of a classifier in a problem with class imbalance.

Results for Static Sampling. We randomly generated 100 seeds; 50 were used as the seeds for pseudo-random number generators to generate samples from the training dataset and the remaining 50 were used to generate, using ramped half-and-half, the initial populations of GP trees. Using the Yeast_{MIT} dataset and 50 as the sample size, we ran the GP search process for each combination of sample and initial tree population.

Table 2. Results of standard GP approach, with standard fitness function over the five datasets, mean $\pm$ standard deviation over 50 runs

Data Set	Overall Accuracy	Minority (%)	Majority (%)
Yeast_{POX}	99.2 ± 0.2	45.0 ± 16.2	99.9 ± 0.1
Yeast_{ME1}	97.2 ± 0.5	36.8 ± 19.2	99.1 ± 0.6
Yeast_{MIT}	84.6 ± 1.0	39.2 ± 7.8	93.5 ± 2.1
SPECT	78.0 ± 2.0	$\mathbf{4.1 \pm 10.7}$	96.6 ± 3.0
Pima	66.8 ± 2.1	32.0 ± 8.9	85.4 ± 5.0

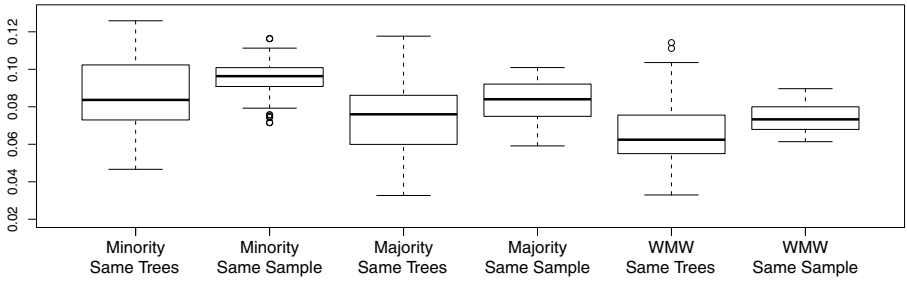


Fig. 1. Results showing *variability* of performance using static sampling. Boxplot of comparison of standard deviations of classification accuracy of GP (the vertical axis) using the same trees with GP using the same samples, over the Minority and Majority Classes and WMW Statistic.

For each initial population of trees there were 50 runs, each with a different sample used for evaluation during evolution. Similarly, for each sample there were 50 GP runs, each with a different initial population of trees. For each of these we calculated the standard deviation of classification accuracy of the minority and majority classes, and the Wilcoxon-Mann-Whitney statistic. The boxplots in Figure 1 provide comparison between the distributions of standard deviations across GP with the same sample and different trees, and GP with different samples but the same initial population of trees. It is shown that the median of classification accuracy on the majority class, the minority class and the WMW is higher when the sample is the same than when the initial population of trees is the same. It is also evident the the spread of the standard deviations is greater when taken across runs with the same initial populations of trees than the same sample. Over minority and majority classes and the WMW the variation from different samples with the same trees is comparable to the variation that comes from the differing initial trees (with the sample sample). There is a slightly greater spread across samples with same tree than trees with the same sample. This shows that the variation introduced by taking different samples is no worse than the variation between GP runs already accepted.

Results for Dynamic Sampling. For the experiments using sampling we used the same evolutionary parameters as with standard GP. We generated 50 random numbers to act as the seeds for the random number generators in each of the 50 GP runs for each of the six sampling methods. Accuracy of classification of the test data is measured using three statistics: the average performance over each class, i.e., $\frac{hits_c}{N_c}$, and the Wilcoxon-Mann-Whitney (WMW) statistic. Table 3 presents the results of the sampling methods on our five datasets, in the form of the three statistics above and their standard deviations. The average run time of a evolutionary run is also given (in seconds).

Table 3 shows that all sampling methods improved the classification accuracy of the minority class and decreased the classification accuracy of the majority class in comparison to Standard GP. Basic Under-sampling has the fastest

Table 3. Results of GP with sampling over the test sets of the five datasets with non-dominated results in bold

Sampling Method	Overall (%)	Minority (%)	Majority (%)	WMW Statistic	Run time (s)
Yeast$_{POX}$					
Basic Under	74.47	**72.0 ± 13.4**	**74.5 ± 13.0**	0.67 ± 0.13	4.0
Under A	88.81	53.2 ± 22.2	89.3 ± 9.1	0.51 ± 0.22	314.1
Under B	99.20	**55.4 ± 18.1**	**99.8 ± 0.4**	0.55 ± 0.18	105.1
Basic Over	87.22	59.4 ± 16.5	87.6 ± 8.8	0.57 ± 0.16	214.0
Over A	90.13	55.8 ± 22.0	90.6 ± 9.1	0.53 ± 0.21	306.0
Over B	97.89	**60.2 ± 19.6**	**98.4 ± 3.7**	0.60 ± 0.20	115.9
Yeast$_{ME1}$					
Basic Under	92.53	**96.9 ± 4.7**	**92.4 ± 2.9**	0.95 ± 0.05	0.6
Under A	96.40	**90.0 ± 7.3**	**96.6 ± 1.0**	0.89 ± 0.07	114.4
Under B	95.70	92.5 ± 5.3	95.8 ± 1.4	0.91 ± 0.05	136.3
Basic Over	96.50	**89.8 ± 6.8**	**96.7 ± 1.1**	0.89 ± 0.07	178.8
Over A	96.55	**88.5 ± 6.0**	**96.8 ± 1.2**	0.87 ± 0.06	110.5
Over B	96.06	**94.9 ± 6.1**	**96.1 ± 0.9**	0.94 ± 0.06	140.1
Yeast$_{MIT}$					
Basic Under	78.22	**62.6 ± 6.6**	**81.3 ± 4.7**	0.58 ± 0.05	33.8
Under A	79.90	61.1 ± 4.6	83.6 ± 3.1	0.57 ± 0.04	181.4
Under B	79.72	60.0 ± 5.5	83.6 ± 3.6	0.56 ± 0.05	189.7
Basic Over	78.51	**61.8 ± 7.0**	**81.8 ± 4.9**	0.57 ± 0.05	174.1
Over A	80.52	**61.3 ± 5.2**	**84.3 ± 2.8**	0.58 ± 0.04	181.9
Over B	79.33	60.7 ± 6.3	83.0 ± 4.5	0.56 ± 0.04	202.1
SPECT					
Basic Under	75.18	62.4 ± 10.2	78.4 ± 4.8	0.51 ± 0.07	9.2
Under A	76.85	**61.6 ± 9.9**	**80.7 ± 4.0**	0.51 ± 0.07	21.0
Under B	76.33	**61.8 ± 11.3**	**80.0 ± 4.7**	0.51 ± 0.08	22.6
Basic Over	77.16	**56.4 ± 8.2**	**82.4 ± 4.0**	0.48 ± 0.06	24.7
Over A	75.85	61.0 ± 10.5	79.6 ± 5.1	0.51 ± 0.07	21.4
Over B	76.41	**63.0 ± 10.9**	**79.8 ± 4.8**	0.52 ± 0.07	23.3
Pima					
Basic Under	62.79	**62.2 ± 10.0**	**63.1 ± 8.1**	0.52 ± 0.07	36.3
Under A	59.60	61.1 ± 6.6	58.8 ± 5.3	0.49 ± 0.06	68.7
Under B	60.78	62.6 ± 6.7	59.8 ± 7.7	0.50 ± 0.06	72.0
Basic Over	63.08	**63.6 ± 8.9**	**62.8 ± 6.4**	0.54 ± 0.07	70.3
Over A	59.89	61.0 ± 6.7	59.3 ± 7.0	0.49 ± 0.05	65.9
Over B	60.08	63.4 ± 5.9	58.3 ± 6.0	0.51 ± 0.05	76.9

run time of all the sampling methods, and also had the best average minority class and lowest majority class classification accuracies (except for Pima). Basic Under-sampling has the highest WMW accuracy and also gave the smallest difference in class accuracy. Over-sampling B has the next highest minority class accuracy and WMW. Basic Under-sampling achieved the best results for the three datasets Yeast$_{POX}$, Yeast$_{ME1}$ and Yeast$_{MIT}$. Over-sampling B gives a more balanced performance between the minority and majority classes on the Yeast$_{ME1}$ data, but is less accurate at classifying the minority class.

Yeast$_{POX}$ has a large standard deviation in the minority class under all of the sampling methods. This can be partly explained by the few minority class instances, only 10 in each of the training and test sets. The highest WMW and minority class classification accuracy on the SPECT dataset was with Over-sampling B. The Pima dataset had highest minority class classification accuracy under Basic Over-sampling, closely followed by Over-sampling B. The classification accuracy was very balanced across the minority and majority classes.

Over all datasets but Pima Basic Under-sampling performed better than Basic Over-sampling on the minority class and worse on the majority class. Under-sampling A is the only sampling method which does not give the best classification accuracy on either majority or minority class for one of the five datasets.

Robustness and the Pareto Front. In order to evaluate the robustness of the classifiers evolved through these sampling methods we used standard GP with a modified fitness function.

$$fitness = \alpha \left(\frac{hits_{min}}{N_{min}} \right) + (1 - \alpha) \left(\frac{hits_{maj}}{N_{maj}} \right)$$

where $0 \leq \alpha \leq 1$, *hits* is the number of correct classifications and N_{min} and N_{maj} are the number of minority and majority class instances respectively. Figure 2 has plotted the results for 50 runs for each of nine α values and the six sampling methods. Here, α fitness on the training set approximates the training set Pareto front. For each sampling method there are four points plotted for each dataset, corresponding to the retention of the sample for one, five, 10 and 15 generations. Each sampling method is represented by a different symbol, and the different number of generations by different colours. In each of the six dynamic sampling methods above, a new sample is taken from the training set at each generation and used to evaluate that generation's classification ability. We then modified these methods so that a sample was taken and used to evaluate not only that generation, but the next 4, 9 or 14 generations as well, that is a new sample was taken every 5, 10 or 15 generations during evolution.

The main point to note from the results presented in Figure 2 is that the cluster of results from the six dynamic sampling methods lies *outside* the front comprised of the test-set results for those classifiers on the training-set Pareto front. This clearly demonstrates that more robust classifiers have been evolved by using dynamic sampling. It is much less important that it is difficult to distinguish the constituent results in the cluster.

With standard GP with the α fitness function, Pima Indian Diabetes dataset has a large gap in the classification curve on both test and training datasets. Yeast$_{ME1}$ with standard GP with α fitness function: both minority and majority class accuracy never fall below 45%. The sampling results for the Pima Indian Diabetes dataset are clustered in a very small area. Interestingly, for Yeast$_{POX}$ with standard GP with the α fitness function the classification accuracy on the minority class of the test dataset never falls below 70%. None of the sampling methods are able to improve on the results of when $\alpha = 0.1$. This is probably

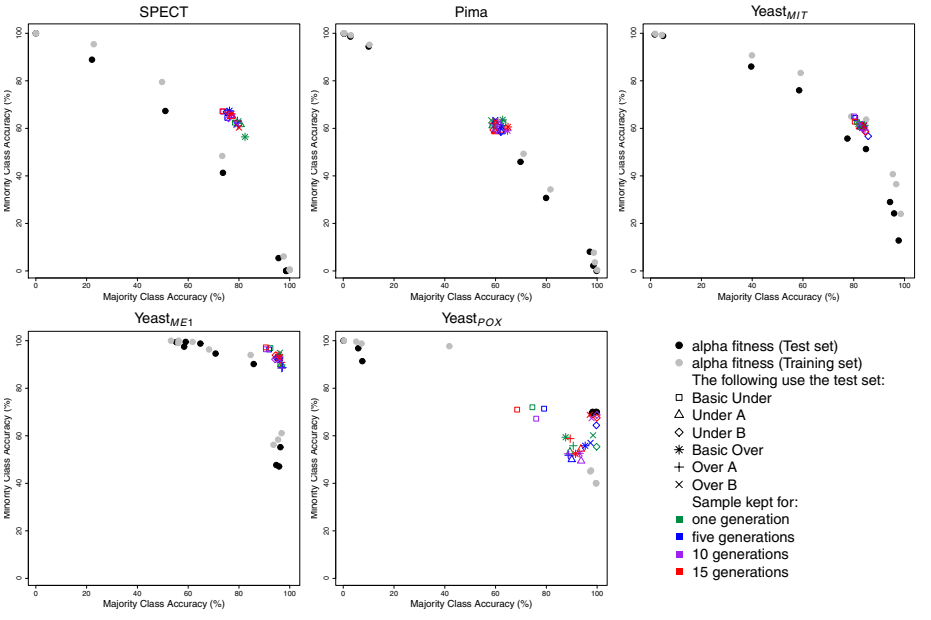


Fig. 2. Results of standard GP with α fitness function for $0 \leq \alpha \leq 1$ across test and training sets, and the six sampling methods

due to the small number of minority class instances that are present in the test dataset (only 10 instances). For Pima, SPECT and Yeast$_{ME1}$ datasets it is clear from Figure 2 that the sample methods increase the classification accuracy beyond the α fitness frontier. This means that the classifiers evolved through the sampling methods are more robust than those from the α fitness function with standard GP.

The impact of retaining the sample for more than one generation varies across datasets and across sampling methods. In most cases the run time decreases substantially when a sample is retained for more than one generation. However, surprisingly, in the case of over-sampling methods A and B the run-time increased.

6 Conclusions

The goal of this work was to address the class-imbalance problem by developing sampling methods to improve the classification performance of genetic programming on unbalanced datasets. By examining the results achieved with our sampling methods in comparison to the results achieved with standard GP it is shown that this was achieved. The methods used do not create balanced classification success rates across the minority and majority classes. Comparing the results achieved in our results to those achieved by Bhowan et al. in [2] shows that the fitness function approach had more success at increasing classification accuracy.

Our sampling methods generate programs with greater robustness due to being evaluated on a changing subset of instances throughout evolution. Further work into the combination of sampling methods with fitness functions also has the potential to increase accuracy of classification. It would be interesting to alter our sampling methods to sample with replacement instead of without replacement, and see what effect this has on the classification accuracy both overall and on the individual classes. In future it would be useful to extend these sampling methods in a Multi-Objective Genetic Programming (MOGP) framework.

Acknowledgment. This work was supported in part by the University Research Fund (URF09-2399/85608) at Victoria University of Wellington and by the Marsden Fund (08-VUW-014) administrated by the Royal Society of New Zealand.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
2. Bhowan, U., Johnston, M., Zhang, M.: Differentiating between individual class performance in genetic programming fitness for classification with unbalanced data. In: CEC 2009: Proceedings of the Eleventh conference on Congress on Evolutionary Computation, pp. 2802–2809 (2009)
3. Doucette, J., Heywood, M.I.: GP classification under imbalanced data sets: active sub-sampling and AUC Approximation. In: O'Neill, M., Vanneschi, L., Gustafson, S., Esparcia Alcázar, A.I., De Falco, I., Della Cioppa, A., Tarantino, E. (eds.) EuroGP 2008. LNCS, vol. 4971, pp. 266–277. Springer, Heidelberg (2008)
4. Gathercole, C., Ross, P.: Dynamic training subset selection for supervised learning in genetic programming. In: PPSN, pp. 312–321 (1994)
5. Gray, H.F., Maxwell, R.J., Martinez-Perez, I., Arus, C., Cerdan, S.: Genetic programming for classification of brain tumours from nuclear magnetic resonance biopsy spectra. In: Koza, J.R., Goldberg, D.E., Fogel, D.B., Riolo, R.L. (eds.) Genetic Programming 1996: Proceedings of the First Annual Conference, p. 424. MIT Press, Stanford University (July 28-31, 1996)
6. Iba, H.: Bagging, boosting, and bloating in Genetic Programming. In: Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M., Smith, R.E. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference, vol. 2, pp. 1053–1060. Morgan Kaufmann, Orlando (July 13-17, 1999)
7. Paris, G., Robilliard, D., Fonlupt, C.: Applying boosting techniques to genetic programming. In: Selected Papers from the 5th European Conference on Artificial Evolution, pp. 267–280. Springer, London (2002)
8. Song, D., Heywood, M.I., Zincir-Heywood, A.N.: A linear genetic programming approach to intrusion detection. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 2325–2336 (2003)
9. Yan, L., Dodier, R.H., Mozer, M., Wolniewicz, R.H.: Optimizing classifier performance via an approximation to the Wilcoxon-Mann-Whitney Statistic. In: International Conference on Machine Learning, pp. 848–855 (2003)

Evolutionary Non Photo–Realistic Animations with Triangular Brushstrokes

Ashkan Izadi[1], Vic Ciesielski[1], and Marsha Berry[2]

[1] School of Computer Science and Information Technology
[2] School of Media and Communication,
RMIT University,
Melbourne, 3000, VIC, Australia

Abstract. We have developed a method for generating non photorealistic animations of a target image. The animations start as a random collection of triangular strokes on a canvas and the target gradually emerges as the animation proceeds. We use genetic programming to evolve programs that draw the brushstrokes. A measure of similarity to the target is used as the fitness function. The best individual in a generation becomes a frame of the animation. We have experimented with *open* and *filled triangles*. Both kinds of triangles resulted in animations that our artist collaborators found engaging and interesting. In particular, the use of filled triangles generated animations that exhibited a novel immersive quality. The evolutionary approach requires artistic judgment in selecting the target images and values for the various parameters and provides a rich environment for exploring novel non photo–realistic renderings.

Keywords: Artificial intelligence, Computer graphic, Evolutionary art, Evolutionary computation, Genetic programming.

1 Introduction

It has been claimed that one of the major benefits of using evolutionary computing to solve an art or design problem is that the evolutionary process can generate novel, unlikely combinations that a human designer or artist would not think of [4]. Non photorealistic rendering (NPR) is an area in which artists are continually seeking novel, creative and unusual kinds of renderings. Our motivation for the work presented in this paper is to investigate whether it might be possible to find a new style of non photorealistic rendering using a form of evolutionary computing.

Non photorealistic rendering is a developing art-form where the goal is to show an image in some more interesting way than as a photograph. Some examples of non photorealistic renderings include pencil sketches, oil paintings, watercolour paintings and photomosaics. Traditionally non photorealistic renderings are still images. However, the use of evolutionary techniques provides an opportunity to add the time dimension. In an evolutionary run the best individual in a generation gets fitter and fitter as the generations increase. If the measure of fitness is similarity to a target image, then the best individual will become more and

J. Li (Ed.): AI 2010, LNAI 6464, pp. 283–292, 2010.

more like the target image as the generations increase. We have found that the best individuals can be combined into an engaging movie that holds the viewer's interest as a recognisable subject slowly emerges from a random collection of strokes.

Current evolutionary algorithms are primarily designed for optimisation problems where the aim is to find the optimal solution as quickly as possible with minimal use of resources. These algorithms need some modification for the art domain where the aim is to find interesting images and animations. For example, an animation that reveals the target too quickly in the first few generations will not be very engaging and a rendering that has optimal fitness will be an exact match to the target and not very interesting as a non photorealistic rendering.

The work presented in this paper is based on the work of Barile and et al. [3]. This work was limited to brushstrokes that were simple grey level lines and stroke placement was limited to blending pixel values when two strokes crossed. Nevertheless the system was able to produce animations and still images that excited our collaborating artists. We build on the work of Barile et al. in three ways: (1) We use color brush strokes, not just grey scale, (2) We use more complex triangular brush strokes, not just lines, (3) We explore a wider range of options when a new stroke will overwrite a pixel already drawn by a previous stroke, not just blending. The new options are *partial stroke* where the new stroke is immediately terminated if it will overwrite a previously drawn pixel, and *no stroke* where the new stroke is not drawn at all if it would overwrite a previously drawn pixel. Our overall goal in this work is to explore a range of brush stokes and placement strategies and determine whether they can be used to produce engaging and interesting animations. In particular, our research questions are:

1. How can we construct a system for non photorealistic rendering with triangular brushstrokes using genetic programming?
2. What are the differences between using open and filled triangular strokes?
3. How do the different placement modes (*blend, partial stroke, no stroke*) affect the evolved animations?

Where it is necessary to make aesthetic judgments about evolved artworks we will seek the views our collaborating artists.[1]

2 Related Work

2.1 Evolutionary Art

Dawkins was one of the first researchers to show how evolution on a computer could be combined with the aesthetic preferences of a user to produce pleasing or interesting forms in 1986 [5]. In 1991, Sims [11] produced his animation Panspermia which was about forests of synthetic 3D plant structures. He employed artificial evolution to evolve the growth rules of plants, for example, how

[1] We thank Karen Trist, from Media and Communication, RMIT, for her comments.

fast branches of a plant can grow. In Turbulence and a number of follow up works, McCormack [8] used an evolutionary approach to generate a series of artworks of an imaginary natural world.

2.2 Non Photo-Realistic Rendering

Gooch et al. [6] identify two main approaches to image painting and drawing in NPR. The first approach uses user-assisted painting programs to place brushstrokes on the canvas manually. The second approach uses automated programs to place brushstrokes on the canvas and the stroke positions are calculated without using any user interaction. The automated approach is divided to two different categories. The first category uses standard computer-graphics rendering without any evolutionary computing technique. The second simulates painting with an evolutionary computing techniques.

Evolutionary Approach: Various evolutionary techniques have been used for NPR. For example, Aupetit et al. [1] used an interactive genetic algorithm to evolve parameters for ant paintings. Samet et al. [10] have designed an NPR system based on ant colony optimisation that can navigate and sense the environment of a target image. They generated painterly and pencil sketch renderings. In 2008, Neufeld et al. [9] introduced a system that can replace the user evaluation interaction. This system can produce non photo-realistic image filters as well as normal evolved images. Neufeld et al. employed genetic programming with multi-objective optimization. Barile et al. [2,3] used simple grey line strokes to generate a non photo-realistic rendering by genetic programming.

3 Configuration of Genetic Programming

An overview of the evolutionary process is shown in algorithm 1. Our programs are evolved according to the grammar shown in figure 1. To draw a brushstroke

Algorithm 1. Genetic Programming for Non Photorealistic Rendering

```
Initialise population with four random individuals
while Maximum number of generations is not reached OR pixel differences are too
big (fitness) do
    Generate an image from each individual
    Compute sum of pixel differences with target (fitness)
    If the new best is better than previous best, write frame of movie
    Copy the best individual to new population (elitism)
    Select two individuals for crossover
    Place the children in the new population
    Select one individual for mutation
    Place child in the new population
end while
Compose a movie from individual frames
```

```
NODE = PROGRAM3 | PROGRAM4 | TRIANGLESTROKE
PROGRAM3 = Program3(NODE,NODE,NODE)
PROGRAM4 = Program4(NODE,NODE,NODE,NODE)
TRIANGLESTROKE = Triangle(T,T,T,T,T,T,T,T,T)
T = floating point number
```

Fig. 1. Grammar of Evolved Programs

on the canvas, we use the *triangle* function. This function takes nine floating point arguments as described in in section 3.1. These arguments determine the position, size, orientation and color of the triangle to be drawn. To implement a sequence of instructions in tree based genetic programming requires "gluing" functions [7]. We have used *program3* and *program4* which enable three and four statements respectively to be executed sequentially. These values were determined empirically by experimenting with combinations of values from 2 to 10. Evolved Programs are executed by using pre–order traversal so a tree can be viewed as a linear sequence of brushstrokes. The root of a tree is generally *program3* or *program4*. The *triangle* nodes will always be one level above the leaves, and the leaves can only be the terminals from the *triangle* function. Figure 2 shows an example of an evolved program.

Fitness Function: In our system we use a target image, the fitness of an individual is its similarity to the target as measured by the sum of pixel differences. We sum the differences for the three RGB channels between a pixel on the target image and the corresponding pixel on the evolved image, thus fitness values will decrease over time.

3.1 Open and Closed Triangular Strokes

In the *triangle stroke* function, we have nine terminals, *line length1, line length2, angel1, angel2, x, y, red color, green color* and *blue color*. All the parameters are floating point numbers between 0.0 and 1.0. The line lengths determine the size of the triangles and can be set by artist users. The size of the triangles has a large influence on the style of images that are evolved. We have implemented two

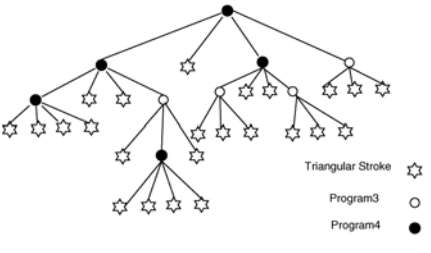


Fig. 2. An evolved program

types of triangular strokes, empty and filled. An empty triangle can be drawn by knowing two lines and the angle between them, and then applying basic geometric calculations for the third line.

3.2 Different Stroke Placement Strategies

There are a number of considerations when placing a stroke on the canvas. Firstly, what to do if the new stroke will write over a pixel that was already written in a previous stroke (placement mode), and secondly whether to use any information from the target image in drawing the stroke (guided or unguided search).

Placement Mode: We have experimented with three different modes for dealing with the situation where a new brushstroke would overwrite a pixel already drawn on the canvas by a previous stroke: *blending, partial stroke, no stroke*. In *blending* mode, if two strokes cross, a new pixel value will be the average of the canvas pixel and the brush pixel. In *partial stroke* mode, the new stroke is drawn until a previously drawn pixel is encountered at which point drawing ceases. In *no stroke* mode, the new stroke is not drawn at all if it would overwrite a previously drawn pixel. In these last two modes, once a pixel has been drawn there is no possibility of improvement with a later brushstrokes.

Guided and Unguided Search: Using information from the target image in placing a stroke can enable accelerated convergence. In *guided search* mode a pixel in the evolved image is only changed if its new value brings it closer to the corresponding target pixel. In *unguided search*, a pixel in the evolved image is unconditionally updated by a brush stroke. In some situations of guided search the acceleration is too fast and the target is evident far too early in the first few generations.

4 Experimental Results

We have experimented[2] with many different images and five to twenty runs for each image. We have explored a large number of combinations of values for selection, population size, elitism, crossover and mutation rates. The best configuration is shown in table 1. Surprisingly, small populations are best for this problem domain.

4.1 Open vs. Closed Triangular Brushstrokes

Figure 3 shows a comparison of filled and unfilled triangle brush strokes for the same target image. Figure 4 shows the progression of fitness values during the course of the evaluations for one run. Low fitness values indicate closer resemblance to the target. Not surprisingly, filled triangles have better fitness and converge towards a solution faster than the empty triangular brush strokes. This is because drawing more pixels in a stroke gives more opportunity to be closer to the target.

[2] Some of our animations can be found at `http://evol-art.cs.rmit.edu.au/ai10`

Table 1. Common GP parameters for our experiments

Functions	*program3,4* and *triangle*
Terminals	float numbers
Maximum evaluations	80000–100,000
Max Tree size	8
Min Tree size	3
Population	4
Crossover	50%
Mutation	25%
Elitism	25%
Selection	Roulette Wheel
Termination	Max. generation reached
Stroke mode	*Blending, partial stroke, no stroke*
Target information	*Guided* and *unguided search*

Artists Evaluations: Our collaborating artists have examined a number of evolutionary runs with filled and unfilled triangles for a variety of target images. They conclude that "The filled triangles provide an extension to the application in a way that the empty triangles do not. The filled triangles offer more possibilities for experimenting with perspective and figure and ground relations".

4.2 Different Size of Triangles

The size of triangles can directly influence the look of evolved images. An example is shown in figure 5. In this figure, small triangles have been rendered by two small line lengths ($L1 = 20$ and $L2 = 30$) and large triangles have been

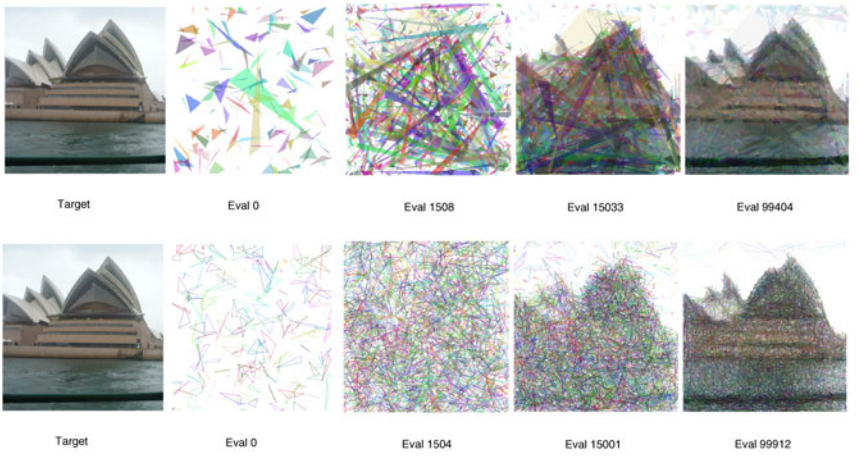


Fig. 3. Sequence of rendered images with the filled triangle (top) and empty triangle (bottom) brushstrokes, blending mode, guided search

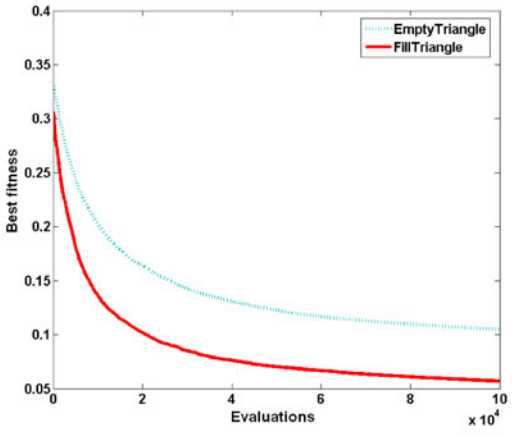


Fig. 4. Fitness progression for the runs of figure 3 for 100,000 evaluations

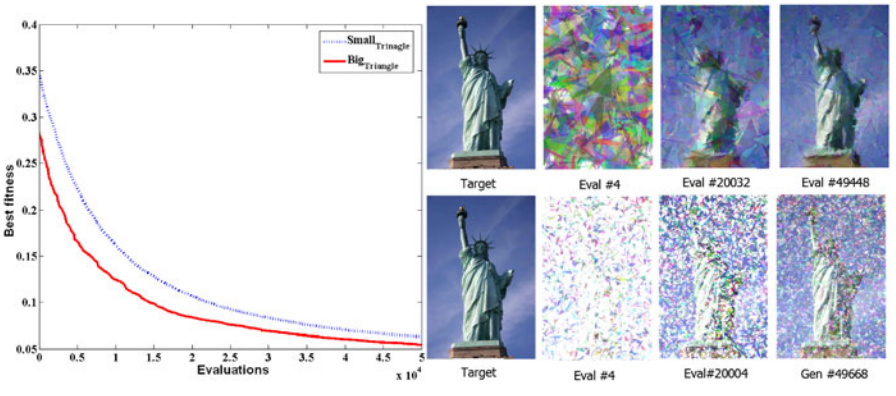


Fig. 5. Left: Fitness for small and large triangles. Right: Rendered images for large and small triangles.

rendered by two large line lengths ($L1 = 150$ and $L2 = 100$). As can be seen from the graph of figure 5, large triangles tend to give faster convergence by providing better fitness values. As can be seen from figure 5 at $50,000$ evaluations, using *blending*, the fitness values are close to each other. However, the final rendered images are quite different. Smaller triangles can give more photorealism, but larger triangles give a more interesting rendering. However, for the *partial stroke* and *no stroke* modes small triangles tend to give faster convergence.

Artists Evaluations: Artists comment that "If the aim is to create an immersive experience that engages the emotions transporting the viewer to an imaginary world as well as an intellectual response, then large triangles with gradients are more likely to satisfy this aim. However, if the intention is to engage analytical

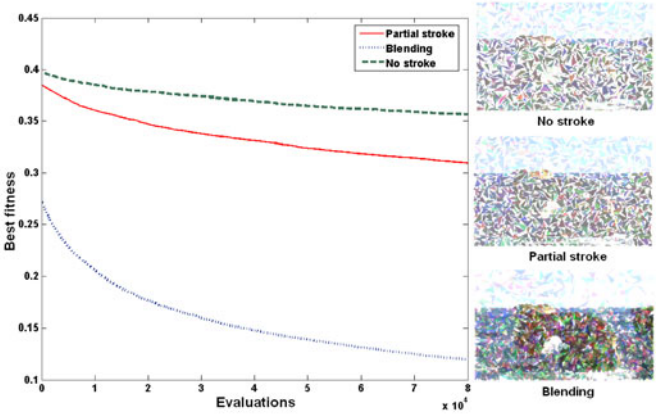


Fig. 6. Comparison of placement modes, unguided search

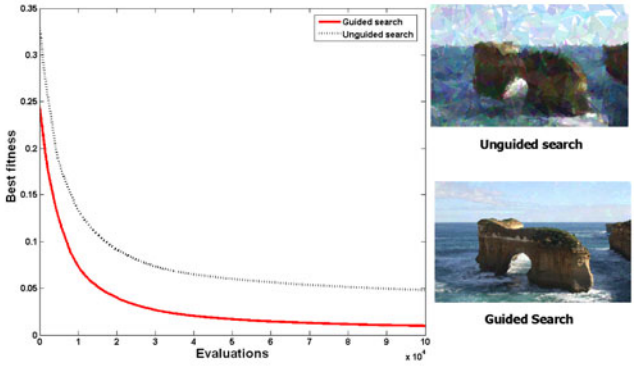


Fig. 7. Left: Fitness differences between guided search and unguided search; Right: unguided search(generation= 45, 589), guided search(generation= 45, 528)

thinking about the image, than smaller triangles may be more appropriate." Also "there is always correlation between thought and feeling and we cannot think without any feeling. Rendering images using a big triangle size causes the images to have a greater affective impact than images made up of small triangles".

4.3 Different Stroke Placement Modes

Unguided Search. Figure 6 shows examples of the three different modes of stroke placement *blending, partial stroke, no stroke*. As can be seen from figure 6, the blending mode converges towards the target much faster than the two other modes. We need more generations to get close to the target with the

"partial stroke" and *"no stroke"* modes. However these two new modes provide different styles of animations and this gives artist users more creative opportunities.

Guided Search. Figure 7 shows the fitness differences between the *"guided search"* and *"unguided search"* approaches. By using *guided search*, the system is able to reach to the target very closely, in some cases, even at the initial generation. *Guided search* over many generations can achieve a high degree of photo-realism. The artist needs to terminate the evolution when the desired degree of photo-realism is achieved.

One current drawback of our algorithms is that they can be slow. A run with guided search for a relatively small target can be done in minutes, while a run with unguided search and a large target can take several days.

5 Conclusion

Our overall goal in this work was to explore a range of brush stokes and placement strategies and determine whether they can be used to produce engaging and interesting animations. We have discovered a number of combinations of triangular brush strokes and placement strategies that can produce animations that artists find interesting, in particular large closed strokes in blending mode. While we cannot claim to have discovered a new style of non photorealistic rendering, we believe that the combination of large triangles and blending is one that a human artist is unlikely to think of.

With respect to our specific research questions, we have shown how to employ genetic programming to generate engaging non photo–realistic animations. This can be done with tree based genetic programming with a *triangle* draw function and two gluing functions, *program3* and *program4*.. We found that a population size four with 50% crossover, 25%mutation and 25% elitism was the best configuration. This is consistent with Barile and et al. [2]. We found that runs with closed triangular brushstrokes converged towards a target faster than open triangular brushstrokes. Our artist collaborators commented that closed triangular brushstrokes create an immersive experience that is more engaging than open triangular brush strokes and that open triangular strokes evoke analytical thinking but filled triangular brushstrokes are more affective. Different modes of stroke placement provide different kinds of renderings. With the *blending* mode, we reach the target quicker and generate a more engaging rendering than with the *partial stroke* and *no stroke* modes. By using information from the target image (guided search), we have shown that fitness is improved around 100% compared with not using any information from the target image (unguided search). Guided search provides better resolution of the final image than unguided search, and images resembling the target are evolved earlier, too early in some cases.

Comments From Our Artist Collaborators: Our artist collaborators made the following comments while working with our programs: (1) "Triangular brush

strokes create some good visual illusions", (2) "Visual illusions made by triangular brush strokes, especially filled triangles, cause an emotional, or affective, response in a viewer. This is a good thing", (3) "I like to have control over variables to create specific effects I have in mind". These comments suggest that our programs are capable of rendering images and animations that are conceptually strong.

Future Work: In future work, we plan to optimize the fitness calculation to speed up the runs, to explore different kinds of brush strokes and to investigate special purpose genetic operators for these kinds of rendering tasks.

References

1. Aupetit, S., Bordeau, V., Monmarche, N., Slimane, M., Venturini, G.: Interactive evolution of ant paintings. In: The 2003 Congress on Evolutionary Computation, CEC 2003, vol. 2, pp. 1376–1383 (8-12, 2003)
2. Barile, P., Ciesielski, V., Berry, M., Trist, K.: Animated drawings rendered by genetic programming. In: GECCO 2009: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, pp. 939–946. ACM Press, New York (2009)
3. Barile, P., Ciesielski, V., Trist, K.: Non-photorealistic rendering using genetic programming. In: Li, X., Kirley, M., Zhang, M., Green, D., Ciesielski, V., Abbass, H.A., Michalewicz, Z., Hendtlass, T., Deb, K., Tan, K.C., Branke, J., Shi, Y. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 299–308. Springer, Heidelberg (2008)
4. Bentley, P., Corne, D.: Creative evolutionary systems. Morgan Kaufmann Pub., San Francisco (2001)
5. Dawkins, R.: The Blind Watchmaker. Longman Scientific & Technical, New York (1986)
6. Gooch, B., Gooch, A.: Non-photorealistic rendering, ch. 4,5,9. AK Peters Wellesley, MA (2001)
7. Koza, J.: Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge (1992)
8. McCormack, J.: Impossible nature: the art of Jon McCormack. Australian Centre for the Moving Image (2004)
9. Neufeld, C., Ross, B.J., Ralph, W.: The evolution of artistic filters. In: The Art of Artificial Evolution, pp. 335–356. Springer, Heidelberg (2008)
10. Semet, Y.O., Reilly, U.M., Durand, F.: An Interactive Artificial Ant Approach to Non-photorealistic Rendering. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 188–200. Springer, Heidelberg (2004)
11. Sims, K.: Artificial evolution for computer graphics. Computer Graphics, 319–328 (1991)

Cellular Differential Evolution Algorithm

Nasimul Noman[1,2] and Hitoshi Iba[1]

[1] University of Tokyo, Tokyo, Japan
{noman,iba}@iba.t.u-tokyo.ac.jp
[2] University of Dhaka, Dhaka, Bangladesh

Abstract. This paper presents a cellular version of Differential Evolution (DE) algorithm. The notion behind the geographical distribution of DE population with local interaction is to study the influence of slow diffusion of information throughout the population. The study was carried out using the compact configuration of neighborhood from which all the auxiliary parents for DE recombination were selected. The empirical study was carried out using a standard benchmark suite consisting of 10 functions. The results show that the structured population with local interaction improves the convergence characteristics of DE and the performance improvement was also verified using scalability study. A brief comparison with cellular GA was also included.

1 Introduction

Evolutionary Algorithms (EAs) may or may not impose some sort of spatial structure on their populations. The large majority of EAs use a single population without any structure and are known as *panmictic* EAs. However, there also exists a tradition of structuring the population in a lattice-like topology, originally for parallel implementation [1]. The most commonly used structured-EA models are *island model* and *diffusion model*. The first one evolves separate subpopulations with occasional migration of individuals inbetween, whereas the latter one maintains overlapping neighborhoods. In diffusion EAs which are also known as *cellular* EAs, genetic interaction for an individual is restricted to a small neighborhood – the spatial structure around that individual [2].

Cellular EAs (cEAs) were originally designed to adapt EAs to fine grained architectures. The essence behind the idea is the effective decentralization of selection algorithms in order to exploit finely grained parallel architecture. Hence, the genetic interaction is restricted to a local neighborhood and thereby reduce the communication cost [3]. Nevertheless, the decentralized population management not only privileges the parallel implementation of the algorithm but also helps to improve the sampling of the search space [4,5]. The small neighborhood with partial overlap in cEAs promotes slow diffusion of information through the grid and thereby has been successful in maintaining the diversity in population [6]. The cEAs' advantage in preventing premature convergence is partly attributed to the lower selection pressure compared to that of panmictic EAs [2].

Differential Evolution (DE) is a very simple to implement EA, which has exhibited better performance than many other EAs [7,8]. Because of its elegant

J. Li (Ed.): AI 2010, LNAI 6464, pp. 293–302, 2010.

features such as simple but powerful search capability, use of few parameters, high convergence characteristics and robustness, DE has found many applications in real-world problems [9,10].

In this work, we present the *cellular* Differential Evolution (cDE) algorithm for global optimization. In literature, a couple of parallel implementations of Differential Evolution algorithm have been reported [11,12,13,14]. But, in all of these proposals, the coarse-grained implementation (i.e. the island model) of DE has been presented where the purpose was mainly to speedup the algorithm by taking the advantage of parallel architecture. In the contrary, this work presents a fine-grained implementation of DE – primarily to study its behavior in local spatial structure rather than to attempt a distributed implementation. And the results show that the diffuse implementation has improved the overall performance of the algorithm.

This paper is organized as follows. In Section 2, we present a brief overview of cellular EA. The canonical DE and the proposed cellular version of DE are presented in section 3 and 4 respectively. Section 5 summarizes our empirical studies on benchmark functions and presents statistical and comparative analysis of results. Finally, Section 6 concludes the paper.

2 Cellular Genetic Algorithm (cGA)

In this section, we present a standard model of cellular Genetic Algorithm (cGA), an important kind of cEA. This model of cGA or its variants have been studied by many other researchers [2,15,6]. In this model, the population is spatially structured in a two-dimensional toroidal grid where each grid-point contains exactly one individual. The neighborhood of an individual is defined by the surrounding grid-points which are accessible by a specific number of steps. If the steps are allowed in axial directions only (north, south, east and west) then the neighborhood shape is called L (linear) and if diagonal steps are allowed as well as then the neighborhood shape is called C (compact) [16]. And a number placed after L or C indicates the number of total individuals (including the central individual) in that neighborhood. Fig. 1 illustrates the concept.

The algorithmic description of cGA is presented in Fig. 2(A) for minimization problems. In contrast to canonical GA, here every individual in the population gets a chance to breed its own offspring mating with other parents chosen from its neighborhood. And the offspring will replace its principal parent in the central grid if it has a superior fitness value.

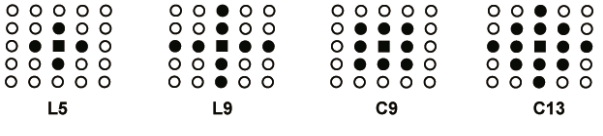


Fig. 1. Neighborhood Models in cellular GA

```
proc cGA
  for i=1 to WIDTH do
    for j=1 to HEIGHT do
      randomly initialize pop(i,j)
      pop(i,j).fitness = Evaluate (pop(i,j))
    end for
  end for
  for s=1 to MAXSTEP do
    for i=1 to WIDTH do
      for j=1 to HEIGHT do
        p1 = pop(i,j)
        p2 =Select(Neighbors(i,j))
        ch = Crossover(p1, p2)
        ch = Mutate(ch)
        ch.fitness = Evaluate (ch)
        if (ch.fitness < pop(i,j).fitness) then
          newPop(i,j) = ch
        else
          newPop(i,j) = pop(i,j)
        end if
      end for
    end for
    pop = newPop
  end for
end proc
```

A) Cellular Genetic Algorithm

```
proc DE
  for i=1 to POPSIZE do
    randomly initialize pop(i)
    pop(i).fitness = Evaluate (pop(i))
  end for
  for s=1 to MAXSTEP do
    for i=1 to POPSIZE do
      p1 = pop(i)
      Select p2, p3 and p4 such that
            p1≠ p2 ≠ p3 ≠ p4
      trial = Mutate(p2, p3, p4)
      ch = Crossover(p1, trial)
      ch.fitness = Evaluate (ch)
      if (ch.fitness < pop(i).fitness) then
        newPop(i) = ch
      else
        newPop(i) = pop(i)
      end if
    end for
    pop = newPop
  end for
end proc
```

B) Canonical Differential Evolution

Fig. 2. Pseudo-code of a simple cGA and canonical DE

3 Differential Evolution (DE)

Differential Evolution (DE) is one of the most elegant new-generation EAs for solving real-parameter optimization problems. Like any other EA, it starts with multiple random initial search points and then systematically guides the population towards global optimum using iterative reproduction and selection [9,17]. The pseudo-code description of canonical DE is presented in Fig. 2(B).

In canonical DE, for each individual, x_G^{p1}, three distinct parents, x_G^{p2}, x_G^{p3} and x_G^{p4}, are selected randomly that participate in a *differential mutation* operation for generating a mutated individual x_G^{trial} as follows:

$$x_G^{trial} = x_G^{p2} + F(x_G^{p3} - x_G^{p4}), \tag{1}$$

where G is the generation number and F, commonly known as *amplification factor*, is a real constant, taken from [0.1, 1.0], that controls the rate at which population evolves [8]. Next, to complement the differential mutation, DE employs a crossover operation, called *discrete recombination*, in which the the genes of offspring x_G^{ch} are inherited from x_G^{p1} and x_G^{trial} determined by a parameter called *crossover probability* ($C_r \in [0, 1]$) as follows:

$$x_{G,j}^{ch} = \begin{cases} x_{G,j}^{trial} & \text{with probability } C_r \\ x_{G,j}^{p1} & \text{with probability } (1 - C_r), \end{cases} \tag{2}$$

where $x_{G,j},\ j = \{1, \cdots, N\}$ denotes j-th parameter of individual vectors. Subsequently, a binary knock-out competition is played between each individual x_G^{p1}, and its offspring x_G^{ch} and the winner is selected deterministically and promoted to next generation. Besides, there are many other variants of DE with different learning strategies about which can be learnt from [9].

4 Cellular Differential Evolution (cDE)

From the description in previous two sections and from Fig. 2, many similarities between cGA and canonical DE become apparent. Both algorithms give every individual a chance to spawn its own offspring irrespective to its fitness and both play a knock-out tournament between every offspring and its parent for survival. The substantial differences between these two algorithms are the presence/absence of spatial structure with the neighborhood concept and the recombination operators with their application order.

Essentially, it will be interesting to observe how DE will behave if fine-grained structure is applied to its population and the current work presents our empirical study in this regard. We propose a spatially structured version of DE, called cellular DE (cDE) which is outlined in Fig. 3. This cellular version of DE is expected to improve the convergence characteristics of the algorithm by exploring the search space using the overlapping neighborhood [15].

```
proc cDE
    for i=1 to WIDTH do
        for j=1 to HEIGHT do
            randomly initialize pop(i,j)
            pop(i,j).fitness = Evaluate (pop(i,j))
        end for
    end for
    for s=1 to MAXSTEP do
        for i=1 to WIDTH do
            for j=1 to HEIGHT do
         p1 = pop(i,j)
         p2, p3, p4 =Select(Neighbors(i,j)) such that p1≠ p2 ≠ p3 ≠ p4
         trial = Mutate (p2, p3, p4)
         ch = Crossover(p1, trial)
         ch.fitness = Evaluate (ch)
         if (ch.fitness < pop(i,j).fitness) then
             newPop(i,j) = ch
         else
             newPop(i,j) = pop(i,j)
         end if
            end for
        end for
        pop = newPop
    end for
end proc
```

Fig. 3. Cellular Differential Evolution

From the discussion in Section 2, it is evident that many different neighborhood shapes and sizes are possible for the proposed cDE algorithm. In this study, we adopted a fairly standard neighborhood C9 (See Fig. 1). For each central parent, three auxiliary parents were chosen randomly from its neighborhood such that they are different. The crossover, mutation and the survival selection operations are kept the same as used in canonical DE (see Section 3).

5 Empirical Study

In this section, we present a numerical study that was carried out to assess the performance of cDE using a test suite consisting of commonly used benchmark functions found in literature. The benchmark functions are *Sphere Model* (f_{sph}), *Generalized Rosenbrock's Function* (f_{ros}), *Generalized Schwefel's Problem 2.26* (f_{sch}), *Ackley's Function* (f_{ack}), *Generalized Griewank Function* (f_{grw}), *Generalized Rastrigin's Function* (f_{ras}), *Salmon's Function* (f_{sal}), *Whitley's Function* (f_{wht}), *Generalized Penalized Function 1* (f_{pn1}) and *Generalized Penalized Function 2* (f_{pn2}). More details about these functions' definition and characteristics can be found in [18] and [9]. Here we study the optimization of this benchmark suite using the proposed cellular version of DE. Although the focus is to compare the proposed cDE with canonical DE, we also studied the competitiveness of cDE comparing with real-valued cGA.

5.1 Experimental Study

We performed 25 independent runs on each algorithm and present the statistics of the trials. Although most of the experiments were done at $N = 30$ dimension, some were done at $N = 50$ and $N = 100$ dimensions to check the effect on scalability. In each dimension, we allowed an algorithm $10000 \times N$ fitness evaluations at maximum to find the global optimum. If the algorithm can reach an *error value*, defined as $(f(x) - f(x^*))$ where x^* is the global optimum of f, less than $\epsilon = 10^{-6}$ within the maximum allowed fitness evaluations then we assume that it has found the global optimum. We present the statistics of error values obtained at the end ($AVG_{Er} \pm SD_{Er}(CNT)$) and the statistics of the fitness evaluations ($AVG_{Ev} \pm SD_{Ev}$) required to reach the global optimum (for successful cases only). CNT denotes the number of successful trials that reach global optimum.

5.2 Experimental Setup

Although the canonical DE needs only three control-parameters, the choice of these parameters is critical for its performance. For classic DE, $F = 0.9$ and $C_r = 0.9$ is a robust setting [10,9]. Hence, we used this fixed setting for parameters in all experiments both in DE and cDE. As we have restricted the maximum number of allowed fitness evaluations the performance of DE is greatly influenced by its population size. For canonical DE and cDE we used $P \approx N$ as recommended in [10]. More specifically, the population size and the grid size for the algorithms were $P = 30$ (6×5), $P = 49$ (7×7) and $P = 100$ (10×10) for $N = 30$, $N = 50$ and $N = 100$ dimensions respectively.

Table 1. Statistics of error values and fitness evaluations for N=30 and Popsize=30

	Best Error Values		Required Fitness Evaluations	
F	DE	cDE	DE	cDE
F_{sph}	5.22E-17±1.44E-16 (25)	**3.94E-21±4.45E-21 (25)**	147067.3±8652.8	**126848.2±4760.8**
F_{ros}	1.29E+01±1.28E+01 (0)	**1.18E+01±1.50E+01 (0)**	–	–
F_{sch}	9.60E-03±8.86E-03 (0)	**1.03E-03±1.73E-03 (0)**	–	–
F_{ack}	8.22E-10±9.84E-10 (25)	**2.03E-11±3.06E-11 (25)**	212332.3±8822.8	**182931.6±7986.9**
F_{grw}	3.74E-03±6.00E-03 (17)	**1.97E-03±3.63E-03 (19)**	103682.2±69208.8	**95707.9±59077.7**
F_{ras}	2.69E+01±7.90E+00 (0)	**2.49E+01±6.47E+00 (0)**	–	–
F_{sal}	2.56E-01±5.02E-02 (0)	**2.34E-01±4.54E-02 (0)**	–	–
F_{wht}	3.50E+02±6.09E+01 (0)	**2.88E+02±1.13E+02 (0)**	–	–
F_{pn1}	4.15E-02±1.20E-01 (22)	**2.07E-02±8.46E-02 (23)**	127959.3±42775.5	**105774.2±34487.2**
F_{pn2}	4.39E-04±2.20E-03 (24)	**5.45E-20±1.59E-19 (25)**	145197.8±34030.8	**128319.6±11263.5**
F_{ros}†	5.03E+00±2.52E+00 (0)	**6.69E-01±1.38E+00 (4)**	-	**57698.5±115397.0**
F_{ras}‡	00E+00±00E+00 (25)	00E+00±00E+00 (25)	180296.0±9812.0	**176798.3 ±9424.8**

† $Popsize = 50$, $C_r = 0.95$, $F = 0.75$ ‡ $Popsize = 50$, $C_r = 0.2$ $F = 0.5$

Table 2. Statistics of error values and fitness evaluations for N=50 and Popsize=49

	Best Error Values		Required Fitness Evaluations	
F	DE	cDE	DE	cDE
F_{sph}	2.95E-03 ± 3.47E-03 (0)	**6.96E-12 ± 7.49E-12 (25)**	–	**334994.5 ± 13242.8**
F_{ros}	2.46E+02 ± 2.54E+02 (0)	**5.74E+01 ± 3.06E+01 (0)**	–	±
F_{sch}	4.93E+03 ± 2.46E+03 (0)	**5.50E+01 ± 2.65E+01 (0)**	–	±
F_{ack}	8.99E-03 ± 3.68E-03 (0)	**5.54E-07 ± 2.28E-07 (23)**	–	**438030.3 ± 138564.0**
F_{grw}	4.28E-03 ± 3.81E-03 (0)	**8.88E-04 ± 2.45E-03 (22)**	–	**308906.2 ± 100816.8**
F_{ras}	5.96E+01 ± 2.19E+01 (0)	**5.10E+01 ± 1.21E+01 (0)**	–	±
F_{sal}	1.14E+00 ± 1.66E-01 (0)	**3.91E-01 ± 5.01E-02 (0)**	–	±
F_{wht}	1.96E+06 ± 9.33E+06 (0)	**7.23E+02 ± 2.71E+02 (0)**	–	±
F_{pn1}	3.05E-02 ± 8.02E-02 (0)	**2.49E-02 ± 6.95E-02 (21)**	–	**278000.3 ± 120315.5**
F_{pn2}	4.77E-02 ± 7.55E-02 (0)	**1.01E-10 ± 1.85E-10 (25)**	–	**361820.1 ± 20777.1**

5.3 Results and Analysis

The results presented in Table 1, show that compared to canonical DE, the proposed cDE algorithm was successful to achieve a lower error value and/or could reach the global optimum using fewer fitness evaluation and/or reached the global optimum in greater number of trials. For some functions, the performance of both DE and cDE may not seem to be the best. This is because we did not perform any parameter tuning and a detailed parameter study is beyond the scope of this work. Just to give a an idea, we tuned the parameters of both algorithms as recommended in [9] for Rosenbrock and Rastrigin functions respectively, and the results are added at the bottom of Table 1, which once again shows the superiority of cDE. These additional results give us more confidence about the success of the cellular version of the algorithm.

In order to test the effect of scalability, we evaluated the algorithms at higher dimensions and the results for N=50 and N=100 are presented in Table 2 and 3 respectively. In every case, the newly proposed cDE algorithm exhibited superior performance compared to the classic DE. In order to validate the presented results we performed some statistical analysis of the results in Table 1, 2 and 3 using Student's t-test and the p-values are reported in Table 4. For N=30 dimension, in most of the cases the t-test indicate significant difference and in higher dimensions, it shows very significant difference in every case.

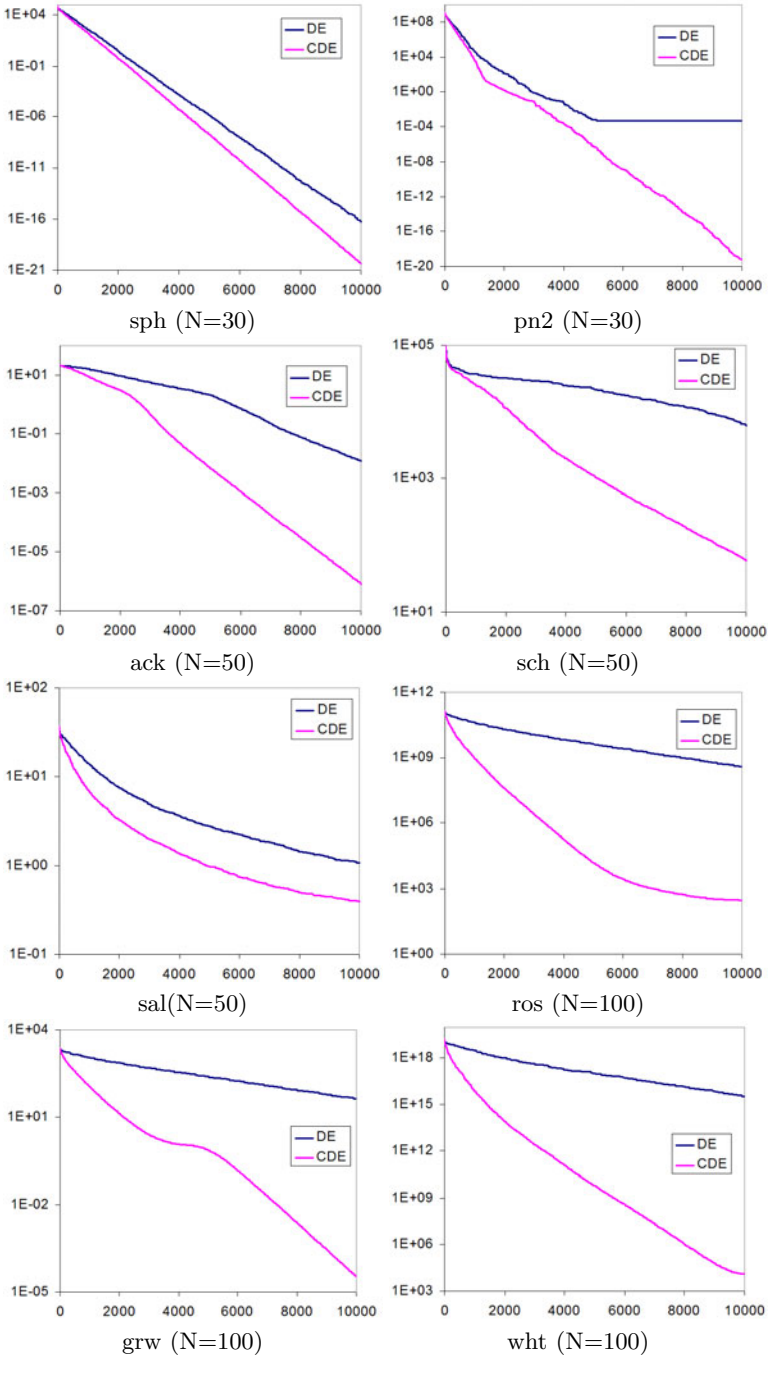


Fig. 4. Convergence graphs for comparing cDE and canonical DE (x axis represents generation and y axis represents fitness)

Table 3. Statistics of error values for N=100 and Popsize=100

F	DE	cDE		DE	cDE
F_{sph}	4.42E+03±1.16E+03	**7.27E-05±3.17E-05**	F_{ras}	8.84E+02±4.39E+01	**1.16E+02±1.88E+01**
F_{ros}	3.82E+08±1.85E+08	**2.86E+02±1.88E+02**	F_{sal}	9.67E+00±1.00E+00	**1.05E+00±1.59E-01**
F_{sch}	1.14E+05±8.26E+03	**5.45E+03±1.37E+03**	F_{wht}	3.49E+15±1.68E+15	**1.18E+04±2.83E+03**
F_{ack}	9.05E+00±7.05E-01	**1.16E-03±3.96E-04**	F_{pn1}	5.11E+05±4.32E+05	**2.92E-02±4.84E-02**
F_{grw}	4.24E+01±9.70E+00	**3.41E-05±2.18E-05**	F_{pn2}	3.49E+06±1.74E+06	**2.34E-01±7.92E-01**

Table 4. p values of t-Distribution calculated from Table 1, Table 2 and Table 3

N=30			**N=50**			**N=100**		
F	p-val (Er)	p-val (Ev)	F	p-val (Er)	p-val (Ev)	F	p-val (Er)	p-val (Ev)
F_{sph}	**1.58E-02**	**8.64E-19**	F_{sph}	**3.84E-07**	-	F_{sph}	**2.63E-30**	-
F_{ros}	6.96E-01	-	F_{ros}	**5.17E-06**	-	F_{ros}	**6.77E-19**	-
F_{sch}	**3.32E-08**	-	F_{sch}	**2.84E-18**	-	F_{sch}	**3.18E-55**	-
F_{ack}	**8.77E-07**	**4.53E-22**	F_{ack}	**7.16E-22**	-	F_{ack}	**6.08E-55**	-
F_{grw}	**8.61E-02**	**6.13E-01**	F_{grw}	**4.38E-06**	-	F_{grw}	**6.19E-33**	-
F_{ras}	1.77E-01	-	F_{ras}	**2.25E-02**	-	F_{ras}	**1.26E-59**	-
F_{sal}	**2.79E-02**	-	F_{sal}	**1.19E-32**	-	F_{sal}	**2.34E-46**	-
F_{wht}	**1.61E-03**	-	F_{wht}	1.51E-01	-	F_{wht}	**4.61E-19**	-
F_{pn1}	3.32E-01	**1.13E-02**	F_{pn1}	7.17E-01	-	F_{pn1}	**1.12E-10**	-
F_{pn2}	1.72E-01	**2.27E-03**	F_{pn2}	**6.46E-05**	-	F_{pn2}	**1.74E-18**	-

To further validate the effect of local neighborhood, we compare the average (of the best individuals in different trials) convergence curves of DE and cDE for different functions in Fig. 4. These graphs clearly show that the cellular model has significantly improved the convergence characteristics of the algorithm and at higher dimensions the improvement becomes even more substantial. So based on the results presented in tables and graphs, we can summarize that in general the notion of local neighborhood has improved the convergence characteristic of DE and the performance difference between the newly proposed cDE and DE becomes more significant at higher dimensions.

Table 5. Comparison with the cGA presented in [6]

	cGA			**cDE**		
F	Mean	Std	Success	Mean	Std	Success
F_{sph}	3.70E-05	1.40E-05	0%	**3.94E-21**	**4.45E-21**	**100.00%**
F_{Elp}	2.09E+00	2.01E+00	0%	**1.41E-20**	**2.77E-20**	**100.00%**
F_{sch}	3.51E+00	1.38E+00	0%	**1.03E-03**	**1.73E-03**	**0.00%**
F_{ack}	1.42E-03	3.25E-04	0%	**2.03E-11**	**3.06E-11**	**100.00%**
F_{grw}	4.26E-03	5.97E-03	0%	**1.97E-03**	**3.63E-03**	**76.00%**
F_{ras}	**1.96E+01**	**5.54E+00**	**0%**	2.49E+01	6.47E+00	0.00%
F_{ros}	3.50E+01	2.58E+01	0%	**1.18E+01**	**1.50E+01**	**0.00%**
F_{wst}	4.50E-01	7.60E-01	0%	**1.36E-01**	**3.64E-01**	**72.00%**
F_{scf}	**5.53E-01**	**2.97E-01**	**0%**	2.90E+00	7.66E-01	0.00%
F_{FMS}	1.58E+01	7.31E+00	0%	**1.30E+00**	**2.49E+00**	**72.00%**

Finally, we compared the performance of cDE with that of the canonical cGA presented in [6]. The comparative results are presented in Table 5, in terms of the benchmark presented in [6]. And in every case cDE has outperformed

canonical cGA by far. These results highlight the high-ranking performance of the proposed algorithm compared to real-valued cGA.

6 Conclusion

In this work, we present a study on how the population structuring with local interaction influences the characteristics of Differential Evolution algorithm. Generally, in EAs faster convergence increases the risk of ending in local optima. However, cGA offers a good tradeoff solution to this problem by promoting slow diffusion of individuals through the grid. This work investigates the effect of lower selection pressure exercised by local structuring on classic DE algorithm and the experimental results show that the diffusion model has struck a greater balance between the DE recombination operators and the knockout selection.

Empirical study using a representative benchmark suite has found that the newly proposed algorithm possesses superior convergence characteristics compared to the classic DE. The convergence curves show that the population structuring has made DE significantly faster without compromising the success ratio. The proposed cDE has outperformed the canonical DE in terms of error value, required evaluation and success ratio in almost every studied function. The effect of scalability was also investigated and the results show that at higher dimensions the performance difference becomes even more significant. Statistical validation of the results were also performed. Competitiveness with the existing cGA was also shown. Additionally, the proposed cDE algorithm itself becomes an elegant candidate to exploit the fine grained parallel architecture.

The results presented here provides some important insights about the effect of local interaction in DE algorithm. These results also suggest that introducing topological strategies to DE, we can have a very efficient algorithm for real valued problems. For future study, there are many things to explore such as the effect of neighborhood shape and size, asynchronous updates policies, analyzing the selection pressure etc.

References

1. Alba, E., Troya, J.M.: A survey of parallel distributed genetic algorithms. Complexity 4(4), 31–52 (1999)
2. Alba, E., Giacobini, M., Tomassini, M., Romero, S.: Comparing synchronous and asynchronous cellular genetic algorithms. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 601–610. Springer, Heidelberg (2002)
3. Jong, K.D., Sarma, J.: On decentralizing selection algorithms. In: Sixth International Conference on Genetic Algorithms ICGA 1995, pp. 17–23 (1995)
4. Alba, E., Troya, J.M.: Improving flexibility and efficiency by adding parallelism to genetic algorithms. Statistics and Computing 12(2), 91–114 (2002)
5. Tang, J., Lim, M.H., Ong, Y.S., Er, M.J.: Parallel memetic algorithm with selective local search for large scale quadratic assignment problems. International Journal of Innovative Computing, Information and Control 2(6), 1399–1416 (2006)

6. Huy, N.Q., Soon, O.Y., Hiot, L.M., Krasnogor, N.: Adaptive cellular memetic algorithms. MIT Evolutionary Computation 17(2), 231–256 (2009)
7. Storn, R., Price, K.V.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)
8. Storn, R.: System design by constraint adaptation and differential evolution. IEEE Transactions on Evolutionary Computation 3(1), 22–34 (1999)
9. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution: A Practical Approach to Global Optimization. Springer, Heidelberg (2005)
10. Noman, N., Iba, H.: Accelerating differential evolution using an adaptive local search. IEEE Transactions on Evolutionary Computation 12(1), 107–125 (2008)
11. Zaharie, D., Petcu, D.: Parallel implementation of multi-population differential evolution. In: 2nd Workshop in Concurrent Information Processing and Computing, CIPC 2003 (2003)
12. Tasoulis, D., Pavlidis, N., Plagianakos, V., Vrahatis, M.: Parallel differential evolution. In: Congress on Evolutionary Computation, CEC 2004, pp. 2023–2029 (2004)
13. Apolloni, J., Leguizamn, G., Garca-Nieto, J., Alba, E.: Island based distributed differential evolution: An experimental study on hybrid testbeds. In: Eighth International Conference on Hybrid Intelligent Systems, HIS 2008, pp. 696–701 (2008)
14. Weber, M., Neri, F., Tirronen, V.: Distributed differential evolution with explorative—exploitative population families. Genetic Programming and Evolvable Machines 10(4), 343–371 (2009)
15. Alba, E., Troya, J.M.: Cellular evolutionary algorithms: Evaluating the influence of ratio. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 29–38. Springer, Heidelberg (2000)
16. Sarma, J., Jong, K.D.: An analysis of the effects of neighborhood size and shape on local selection algorithms. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 236–244. Springer, Heidelberg (1996)
17. Noman, N., Iba, H.: A new generation alternation model for differential evolution. In: Genetic and Evolutionary Computation Conference, pp. 1265–1272 (2006)
18. Yao, X., Liu, Y., Liu, G.: Evolutionary programming made faster. IEEE Trans. on Evolutionary Computation 3(2), 82–102 (1999)

A Comparative Study of CMA-ES on Large Scale Global Optimisation

Mohammad Nabi Omidvar and Xiaodong Li

Evolutionary Computing and Machine Learning Laboratory (ECML Lab),
Royal Melbourne Institute of Technology (RMIT), Melbourne, Australia
momidvar@cs.rmit.edu.au, xiaodong.li@rmit.edu.au
http://goanna.cs.rmit.edu.au/~xiaodong/ecml

Abstract. In this paper, we investigate the performance of CMA-ES on large scale non-separable optimisation problems. CMA-ES is a robust local optimiser that has shown great performance on small-scale non-separable optimisation problems. Self-adaptation of a covariance matrix makes it rotational invariant which is a desirable property, especially for solving non-separable problems. The focus of this paper is to compare the performance of CMA-ES with Cooperative Co-evolutionary Algorithms (CCEAs) for large scale global optimisation (on problems with up to 1000 real-valued variables). Since the original CMA-ES is incapable of handling problems with more than several hundreds dimensions, sep-CMA-ES was developed using only the diagonal elements of the covariance matrix. In this paper sep-CMA-ES is compared with several existing CCEAs. Experimental results revealed that the performance of sep-CMA-ES drops significantly when the dimensionality of the problem increases. However, our results suggest that the rotational invariant property of CMA-ES can be utilised in conjunction with a CCEA to further enhance its capability to handle large scale optimisation problems.

1 Introduction

Advances in science and technology provides us with ever more options and features, however having more features makes it more difficult to find the optimum configuration of these decision variables. The rapid growth in the number of decision variables brings a grand scale challenge to optimisation techniques. In nano-technology, the properties of thousands of atoms have to be taken into account in order to produce a substance with a certain property. In aerodynamics, tens of thousands of parameters have to be tweaked in order to optimise a component of a space shuttle to a target shape. This shift in the scale of optimisation problems demands new optimisation techniques capable of dealing with thousands of decision variables.

Many Evolutionary Algorithms (EAs) [6], [3], [2] have been applied to optimisation problems, however the performance of these algorithms, like most of traditional algorithms, deteriorate as the dimensionality of the problem increases. This is referred to as the *curse of dimensionality* [4]. Divide-and-conquer

J. Li (Ed.): AI 2010, LNAI 6464, pp. 303–312, 2010.

is a natural approach for tackling large scale problems. Cooperative Co-evolution (CC) [15] is such a technique that decomposes a large scale problem into a set of sub-problems, each of which is optimised using a separate EA. In the original CC decomposition strategy, each variable is placed in a separate subcomponent. This new paradigm has shown great success on many optimisation problems [15]. However further investigation revealed that this CC decomposition strategy is only effective when there is no interdependency between the decision variables [22]. Shi et al. [19] proposed another technique in which the decision variables are divided into halves. This dividing-in-half strategy does not scale properly as the dimensionality of problem increases, mainly because the size of sub-problems will go beyond the optimisation capabilities of subcomponent optimisers. van den Bergh and Engelbrecht [22] proposed a new decomposition strategy where they decomposed a n-dimensional problem into m s-dimensional subcomponents. It has been shown that this new technique performs better than the original decomposition strategy when dealing with non-separable problems.

Although grouping interacting variables in a common subcomponent increases the performance of many CCEAs to a great extent, capturing the hidden dependencies between the variables is a great challenge by itself. *Random Grouping* (DECC-G, MLCC) [23] and *Delta Grouping* [14] are two major techniques that were proposed recently for capturing the interacting variables. Delta grouping in particular has shown superior performance in capturing interacting variables in grouping them in a common subcomponent [14].

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [10] is a successful optimisation algorithm that has been designed specifically for local optimisation, but it has also shown competitive results for global optimisation [8], [7]. The self-adaptation of covariance matrix makes CMA-ES rotational invariant that makes it a perfect choice for non-separable problems.

Most studies on CMA-ES have been carried out with functions with up to 200 dimensions. There are limited studies in which functions with up to 1000 dimensions have been used [17], however these studies were confined with very few simple test functions. Furthermore, since the introduction of the new test functions such as CEC'2008 [21] and CEC'2010 [20], which are specifically designed for benchmarking of large scale optimisation algorithms, no systematic studies have been carried out on evaluating CMA-ES using these newly proposed functions. In this research we benchmark the performance of CMA-ES on large scale problems proposed in CEC'2008 Special Session and Competition on Large Scale Global Optimisation [21]. In particular we have the following research objectives:

- Benchmarking the performance of standard CMA-ES on large scale problems.
- Comparing the performance of cooperative co-evolutionary algorithms for large scale optimisation with CMA-ES on the same problem set.
- Verifying the performance of a rotational invariant algorithm such as CMA-ES on large scale non-separable optimisation problems.

The organisation of the rest of this paper is as follows. Section 2 explains the preliminaries, such as CC and CMA-ES. Section 3 presents the experimental results and their analysis. Finally, Section 4 summarises this paper and gives directions to future works.

2 Background

This section is dedicated to background information. Section 2.1 describes the Cooperative Co-evolution [15] and different decomposition strategies proposed in the past. Section 2.2 describes CMA-ES [10] in more details.

2.1 Cooperative Co-evolution

Cooperative Co-evolution has been proposed by Potter and De Jong [15], explicitly introduces the notion of modularity in EAs. This notion of modularity is essential in order to solve complex problems. CC works in the form of co-adapted subcomponents. In the context of an optimisation problem, a n-dimensional problem is divided into n 1-dimensional problems, each of which is optimised using a separate EA in a round-robin fashion. This decomposition scheme works well only when there is no interaction between the decision variables. It has been shown that the performance of original CC framework deteriorates when there are interactions between the decision variables [12]. van den Bergh and Engelbrecht [22] used a different decomposition strategy in which they divided a n-dimensional problem into m s-dimensional subproblems. It has been shown that this new decomposition strategy has a better performance compared to the original CC framework [22].

CC framework has been applied to many EAs for large scale optimisation problems. However, the performance of these algorithms deteriorate as the dimensionality of the problem increases. Fast Evolution Programming with Cooperative Co-evolution (FEPCC) [12] is one of the early techniques that has been applied to problems with up to 1000 dimensions. The experimental results revealed that FEPCC performed poorly on one of non-separable problems [12]. The reason for poor performance of FEPCC on such problems is due to grouping of interacting variables in separate subcomponents. As a result, to increase the performance of CCEAs on large scale non-separable problems, the interacting variables have to be identified and be grouped in a common subcomponent.

Random grouping has been proposed by Yang et al. [23] in order to increase the probability of grouping interacting variables in a subcomponent. Although random grouping has shown better performance compared to other algorithms, its performance degrades as the number of interacting variables increases [13]. In order to cope with the increased number of interacting variables, more intelligent and systematic techniques are required to capture interacting variables. Ray and Yao [16] calculated a correlation coefficient matrix from the current population and divided the decision variables into two subcomponents, based on a threshold value on correlation coefficients. This technique does not scale

properly when the dimensionality of the problem increases. This is because the decision variables are divided into halves and as the dimensionality of the problem increases the complexity of sub-problems will go beyond the capabilities of CCEA-AVP. Another disadvantage of CCEA-AVP is the use of correlation coefficients as a measurement for degree of separability (or non-separability) of decision variables. Correlation coefficient measures such as Pearson's correlation matrix measures the linear dependence between two variables which is not a proper estimation for separability (or non-separability) of decision variables.

Delta grouping has been proposed by Omidvar et al. [14] as the first systematic technique for capturing interacting variables. Delta grouping is inspired by the idea of improvement interval under coordinate rotation explained in detailed in [18]. In delta grouping, the average amount of change in every dimension is measured between two consecutive cycles to form a delta vector. Then the variables are sorted based on the absolute magnitude of their delta values. The motivation behind delta grouping is that, in a non-separable problem, when a small delta value is observed in one of the dimensions, there is a high probability to find another decision variable with relatively small delta value. As a result, grouping the variables based on the magnitude of their delta values increases the probability of grouping two interacting variables in one subcomponent. Delta grouping has shown great performance [14] on a wide range of benchmark functions [20], [21] that were proposed especially for large scale global optimisation.

2.2 Covariance Matrix Adaptation Evolution Strategy

CMA-ES is based on Evolution Strategy (ES) [5]. ES is a type of EA which has been extensively used for continuous optimisation tasks. The individuals in ES are real-valued vectors that are systematically changed to get better individuals. Like many EAs, ES rely on three major operations, mutation, recombination, and selection. Mutation and recombination are used for exploration of the search space and generating genetic variations, while the selection operator is for exploitation and convergence to a solution. The mutation operator is an important operator in ES and is central to understanding of CMA-ES. A detailed explanation of various selection operators can be found in [5]. Recombination is not a very common operator in state-of-the-art ES implementations, as a result the focus of this section is on mutation.

Mutation. Mutation is a key operator in ES that generates the maximum genetic variations. In real-valued continuous optimisation problems the mutation is done using a multivariate Gaussian distribution. Equation (1) shows how an individual is mutated using a Gaussian distribution.

$$\tilde{\boldsymbol{y}} = \boldsymbol{y} + \boldsymbol{z} \ , \tag{1}$$

where $\boldsymbol{y}$ is the parent, $\tilde{\boldsymbol{y}}$ is the mutant and the $\boldsymbol{z}$ is defined as follows:

$$\boldsymbol{z} = \sigma\big(N_1(0,1), ..., N_N(0,1)\big) \ , \tag{2}$$

where $N_i(0,1)$ are mutually independent random numbers, generated form a normal distribution with mean zero and standard deviation 1. σ is the strategy parameter which is called the step size. Having only one step size creates an isotropic Gaussian distribution which is symmetric about the mean value which is $\boldsymbol{y}$ in this case. This situation is depicted in Figure 1(a). Having only one step size is not very efficient in solving high dimensional problems. In an extended version of ES, instead of having only one global step size, a vector of step sizes is maintained. Each of the elements in this vector corresponds to one of the dimensions. This allows having different step sizes for every dimension. In this new scheme the mutation is performed using Equations (3).

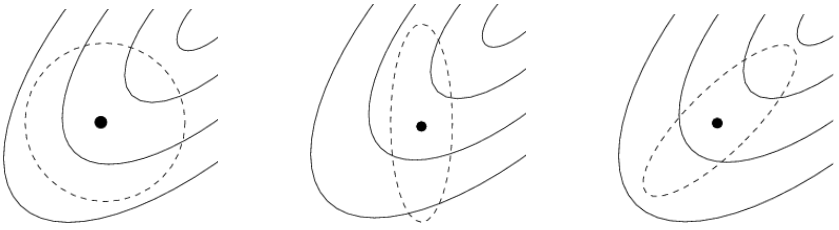
$$\begin{aligned} \boldsymbol{z} &= \big(\sigma_1 N_1(0,1), ..., \sigma_N N_N(0,1)\big) \\ &= D\big(N_1(0,1), ..., N_N(0,1)\big)^T \\ &= DN(0,I) \ , \end{aligned} \tag{3}$$

as it can be seen form Equation (3), there are different σ values for different dimensions. This situation is depicted in Figure 1(b). D is a diagonal matrix with the σ values on the main diagonal. Although this new technique is far more flexible than the isotropic version, it still loses its efficiency when applied to non-separable functions. As it can be seen from Figure 1(b), the Gaussian distribution is scaled in the direction of the coordinate axes. In many real-world problems, the fitness landscape is not aligned with the coordinate system which makes this mutation strategy ineffective. As a result another mutation scheme is needed to work under arbitrary rotations of fitness landscape which is a desirable technique especially for non-separable problems. Covariance Matrix Adaptation ES proposes such a rotational invariant version of ES by self-adapting a rotation matrix M to align the diagonal matrix D with the principal axes of the fitness landscape [10]. So, the mutation scheme in CMA-ES is as follows:

$$\begin{aligned} \boldsymbol{z} &= M\big(\sigma_1 N_1(0,1), ..., \sigma_N N_N(0,1)\big) \\ &= MD\big(N_1(0,1), ..., N_N(0,1)\big)^T \\ &= MDN(0,I) \ , \end{aligned} \tag{4}$$

it is clear that the rotation matrix M creates correlation between the components of $\boldsymbol{z}$, thus, $C = M^T M$. The use of correlation matrix C and the effect of this new mutation scheme is depicted in Figure 1(c). As it can be seen from Figure 1(c) the step sizes are oriented towards the optimum point which is a desirable property for solving non-separable problems.

Adaptation of the Covariance Matrix. The covariance matrix C that was described in the previous section is calculated based on the changes in the mean values of two successive generations. In this case, it is assumed that the current population contains enough information to successfully estimate the correlations. Equation (5) shows how the covariance matrix is updated based on the changes in two successive generations.



(a) $N(0, \sigma^2 I)$ (b) $N(0, D^2)$ (c) $N(0, C)$

Fig. 1. Figure (a): an isotropic distribution is created when there is only one strategy parameter. Figure (b): shows the situation when there is a separate step size for each of the coordinates. Figure (c): is the case where the coordinate system is rotated by a rotation matrix which is derived from the covariance matrix C.

$$C_{\lambda}^{(g+1)} = \frac{1}{\lambda} \sum_{i=1}^{\lambda} (\boldsymbol{y}_i^{(g+1)} - \boldsymbol{m}^{(g)})(\boldsymbol{y}_i^{(g+1)} - \boldsymbol{m}^{(g)})^T \ , \tag{5}$$

where $\boldsymbol{m}^{(g)}$ is the weighted average of μ selected points from the sample of λ offspring in generation g.

In order to accurately estimate the covariance matrix, CMA-ES uses a technique called *cumulation* to utilise the information in the evolution path [10]. The idea of cumulation is simple. Instead of calculating the covariance matrix using only two consecutive generations, an archive of changes in the mean values is recorded and the covariance is updated based on this archive. It is clear that the archive contains far more information about the correlations as compared to using information from only two consecutive generations.

The next step after calculation of the covariance matrix is to find the rotation matrix M from the covariance matrix C in order to stretch the multivariate Gaussian distribution in the direction of the global optimum point. This can be achieved by performing an eigen-decomposition of the covariance matrix C to obtain an orthogonal basis for the matrix C. This orthogonal basis is essentially the matrix of eigen-vectors that can be used for transformation of the sample points. This process is shown in Equation (6).

$$\begin{aligned} N(\boldsymbol{m}, C) &= \boldsymbol{m} + N(0, C) \\ &= \boldsymbol{m} + C^{\frac{1}{2}} N(0, I) \\ &= \boldsymbol{m} + MDM^T N(0, I) \\ &= \boldsymbol{m} + MDN(0, I) \ , \end{aligned} \tag{6}$$

as it can be seen, $DN(0, I)$ is the situation described in Equation (3) and M which is the rotation matrix derived from eigen-decomposition of the covariance matrix C.

One disadvantage of CMA-ES is its relatively high time complexity. This is mainly due to self-adaptation of covariance matrix and eigen-decomposition. It

has been shown that the time complexity of calculating and updating the covariance matrix is of order $O(n^3)$. This makes CMA-ES more computationally expensive compared to other EAs. A few techniques have been proposed to reduce the time complexity of CMA-ES [17], [11], [9], among which sep-CMA-Es was proposed [17]. In a sep-CMA-ES, the covariance matrix C is constrained to be diagonal. This reduces the time complexity from $O(n^3)$ to $O(n^2)$ [17]. However this modification compromises the rotational invariant property of CMA-ES. In another implementation, (1+1)-CMA-ES is proposed in which a very small population size is used [11]. It has been shown that (1+1)-CMA-ES has a time complexity of $O(n^2)$. Although the time complexity of (1+1)-CMA-ES is improved, this modification makes it less appropriate for multimodal test functions due to small population size. A restart CMA-ES has been proposed called IPOP-CMA-ES [1] which is more suitable for multimodal test functions, however IPOP-CMA-ES is only effective for up to 50 dimensions [1]. It is noteworthy that in higher dimensions, very large population sizes are required which significantly increases the computational cost of the algorithm.

3 Experimental Results and Analysis

In this section we present the experimental results of running sep-CMA-ES on CEC'2008 benchmark functions [21]. Tables 1, 2, and 3 contain comparative results of different algorithms on the same benchmark functions. The mean of 25 independent runs are recorded and the best performing algorithms are highlighted in bold.

As it can be seen from Table 1, sep-CMA-ES outperformed other algorithms on 2 out of 6 benchmark functions with 100 decision variables. It is interesting that both of these functions (f_2, f_5) are non-separable. It is also noteworthy that all other algorithms are benefiting from a co-evolutionary framework, and yet sep-CMA-ES performed better on non-separable functions. Another interesting observation is that, sep-CMA-ES performed reasonably better than other algorithms, except for DECC-ML, and DECC-DML, on f_3 which is also a non-separable function. The unique characteristic of DECC-ML, and DECC-DML is that both of them use a uniform random number generator for self-adapting subcomponent sizes. Another observation is that sep-CMA-ES performed poorly on multimodal functions such as f_4, and f_7.

Tables 2, and 3 compare the performance of sep-CMA-ES with other algorithms on 500, and 1000 dimensions respectively. It can be seen that sep-CMA-ES does not have the same relative performance on higher dimensions, in fact, sep-CMA-ES has the best performance only on f_3, but CCEAs are better on all other functions. Overall, the experimental results over all dimensions have shown that sep-CMA-ES does not scale properly as the dimensionality of problem increases. According to Tables 1, 2, and 3 most of the CC algorithms outperformed sep-CMA-ES on almost all of the functions and sep-CMA-ES is often placed last especially on problems with 1000 decision variables.

Table 1. Results of CEC'08 Function on 100 dimensions

Function	DECC	DECC-ML	DECC-D	DECC-DML	MLCC	sep-CMA-ES
f_1	**2.7263e-29**	5.7254e-28	2.9283e-29	4.7379e-28	6.8212e-14	3.1918e-24
f_2	5.4471e+01	2.7974e-04	5.2479e+01	**2.4811e-04**	2.5262e+01	1.3202e+01
f_3	1.4244e+02	1.8871e+02	1.4077e+02	1.9233e+02	1.4984e+02	**4.3300e+00**
f_4	5.3370e+01	**0.0000e+00**	5.4444e+01	**0.0000e+00**	4.3883e-13	2.6324e+02
f_5	2.7589e-03	3.6415e-03	8.8753e-04	7.8858e-04	3.4106e-14	**8.8818e-18**
f_6	2.3646e-01	3.3822e-14	1.2270e-01	**3.1548e-14**	1.1141e-13	6.6495e-01
f_7	-9.9413e+02	-1.5476e+03	-9.8976e+02	**-1.5480e+03**	-1.5439e+03	-1.3625e+03

Table 2. Results of CEC'08 Function on 500 dimensions

Function	DECC	DECC-ML	DECC-D	DECC-DML	MLCC	sep-CMA-ES
f_1	**8.0779e-30**	1.6688e-27	3.8370e-29	1.7117e-27	4.2974e-13	4.2256e-22
f_2	4.0904e+01	1.3396e+00	3.8009e+01	**1.0232e+00**	6.6663e+01	4.8619e+01
f_3	6.6822e+02	5.9341e+02	5.6941e+02	6.8292e+02	9.2466e+02	**3.0788e+02**
f_4	1.3114e+02	**0.0000e+00**	1.4631e+02	**0.0000e+00**	1.7933e-11	1.8262e+03
f_5	2.9584e-04	1.4788e-03	2.9584e-04	2.9584e-04	**2.1259e-13**	9.4260e-02
f_6	6.6507e-14	1.2818e-13	**5.9828e-14**	1.2051e-13	5.3433e-13	8.0505e+00
f_7	-5.5707e+03	**-7.4582e+03**	-4.7796e+03	-7.4579e+03	-7.4350e+03	-6.3868e+03

Table 3. Results of CEC'08 Function on 1000 dimensions

Function	DECC	DECC-ML	DECC-D	DECC-DML	MLCC	sep-CMA-ES
f_1	1.2117e-29	5.1750e-28	**1.0097e-29**	3.3391e-27	8.4583e-13	1.2288e-21
f_2	4.2729e+01	**3.4272e+00**	3.8673e+01	5.81133e+00	1.0871e+02	6.5811e+01
f_3	1.2673e+03	1.0990e+03	1.1597e+03	1.22537e+03	1.7986e+03	**7.9644e+02**
f_4	2.4498e+02	**0.0000e+00**	2.7406e+02	**0.0000e+00**	1.3744e-10	4.2148e+03
f_5	2.9584e-04	9.8489e-04	**1.0392e-15**	1.4611e-15	4.1837e-13	3.6758e-02
f_6	1.3117e-13	2.5295e-13	**1.1866e-13**	2.2908e-13	1.0607e-12	1.9632e+01
f_7^*	-1.4339e+04	**-1.4757e+04**	-1.1035e+04	-1.4750e+04	-1.4703e+04	-1.2419e+04

As it was mentioned earlier in Section 2, sep-CMA-ES has a better time and space complexity as compared to CMA-ES [17], and it has been shown that it outperforms CMA-ES on partially separable problems when the dimension is above 100. On the other hand CMA-ES outperforms sep-CMA-ES on fully non-separable problems, however the full calculation of covariance matrix at the heart of CMA-ES substantially increases its computational cost. The experimental results presented in this paper shows that even sep-CMA-ES performed poorly on benchmark functions especially designed for large scale global optimisation such as CEC'2008 benchmark functions. We speculate that a CC implementation of CMA-ES may hold a great promise and has the benefits of both worlds. CMA-ES has shown superior performance on non-separable functions due to its rotational invariant property and CC is an efficient framework for breaking down a large scale problem into more manageable sub-problems. Using CMA-ES, as the subcomponent optimiser in a CC framework brings the rotational invariant property of CMA-ES with the scalability strength of CC together in one algorithm. Since the covariance matrix adaptation only happens after the

full evaluation of all subcomponents in a CC framework, the CMA part happens less frequently, compensating the high computational cost of CMA-ES.

4 Conclusion and Future Works

In this paper we investigated the use of CMA-ES on large scale non-separable problems. CMA-ES is designed as a highly competitive and robust local optimiser. We have shown that the performance of CMA-ES degrades significantly when the dimensionality of the problem increases. This is true for all of EAs, however CMA-ES suffers from the cures of dimensionality more than other CCEA algorithms, since CMA-ES needs a large population size in order to maintain diversity. This is clearly evident from the performance of CMA-ES on multimodal test functions. Having a large population size is not practical in higher dimensions due to exponential growth in computational cost. Eigen-decomposition of the covariance matrix at the heart of CMA-ES is another source of performance degradation on large scale problem.

Despite its shortcomings on large scale problems, CMA-ES remains a competent solution for optimising small to medium-sized non-separable problems. Experimental results in Table 1 confirms this on non-separable functions with 100 decision variables. This property of CMA-ES makes it ideal to be incorporated into a CC framework as a subcomponent optimiser.

In the future, we intend to develop a CC implementation of CMA-ES for further investigation and comparison on large scale non-separable problems, especially the newly proposed CEC'2010 [20] benchmark function.

References

1. Auger, A., Hansen, N.: A restart CMA evolution strategy with increasing population size. In: McKay, B., et al. (eds.) The 2005 IEEE International Congress on Evolutionary Computation (CEC 2005), vol. 2, pp. 1769–1776 (2005)
2. Bäck, T.: Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Dover Books on Mathematics. Oxford University Press, Oxford (1996)
3. Bäck, T., Fogel, D.B., Michalewicz, Z. (eds.): Handbook of Evolutionary Computation. Institute of Physics Publishing, Bristol, and Oxford University Press, New York (1997)
4. Bellman, R.E.: Dynamic Programming. Dover Books on Mathematics. Princeton University Press, Princeton (1957)
5. Beyer, H., Schwefel, H.: Evolution strategies - a comprehensive introduction. Natural Computing 1(1), 3–52 (2002)
6. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading (1989)
7. Hansen, N.: Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In: Workshop Proceedings of the GECCO Genetic and Evolutionary Computation Conference, pp. 2389–2395. ACM, New York (July 2009)

8. Hansen, N., Kern, S.: Evaluating the CMA evolution strategy on multimodal test functions. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 282–291. Springer, Heidelberg (2004)
9. Hansen, N., Muller, S., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolutionary Computation 11(1), 1–18 (2003)
10. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation 9(2), 159–195 (2001)
11. Igel, C., Suttorp, T., Hansen, N.: A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation GECCO, pp. 453–460. ACM, New York (2006)
12. Liu, Y., Yao, X., Zhao, Q., Higuchi, T.: Scaling up fast evolutionary programming with cooperative coevolution. In: Proceedings of Congress on Evolutionary Computation, pp. 1101–1108 (2001)
13. Omidvar, M.N., Li, X., Yang, Z., Yao, X.: Cooperative co-evolution for large scale optimization through more frequent random grouping. In: Proc. of IEEE World Congress on Computational Intelligence (under review) (2010)
14. Omidvar, M.N., Li, X., Yao, X.: Cooperative co-evolution with delta grouping for large scale non-separable function optimization. In: Proc. of IEEE World Congress on Computational Intelligence (under review) (2010)
15. Potter, M.A., De Jong, K.A.: A cooperative coevolutionary approach to function optimization. In: Proceedings of the Third Conference on Parallel Problem Solving from Nature, vol. 2, pp. 249–257 (1994)
16. Ray, T., Yao, X.: A cooperative coevolutionary algorithm with correlation based adaptive variable partitioning. In: Proc. of IEEE Congress on Evolutionary Computation, pp. 983–989 (May 2009)
17. Ros, R., Hansen, N.: A simple modification in cma-es achieving linear time and space complexity. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 296–305. Springer, Heidelberg (2008)
18. Salomon, R.: Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions - a survey of some theoretical and practical aspects of genetic algorithms. BioSystems 39, 263–278 (1995)
19. Shi, Y., Teng, H., Li, Z.: Cooperative co-evolutionary differential evolution for function optimization. In: Proc. of the First International Conference on Natural Computation, pp. 1080–1088 (2005)
20. Tang, K., Li, X., Suganthan, P.N., Yang, Z., Weise, T.: Benchmark functions for the cec 2010 special session and competition on large-scale global optimization. Technical report, Nature Inspired Computation and Applications Laboratory, USTC, China (2009), `http://nical.ustc.edu.cn/cec10ss.php`
21. Tang, K., Yao, X., Suganthan, P.N., MacNish, C., Chen, Y.P., Chen, C.M., Yang, Z.: Benchmark functions for the cec 2008 special session and competition on large scale global optimization. Technical report, Nature Inspired Computation and Applications Laboratory, USTC, China (2007), `http://nical.ustc.edu.cn/cec08ss.php`
22. van den Bergh, F., Engelbrecht, A.P.: A cooperative approach to particle swarm optimization. IEEE Transactions on Evolutionary Computation 8(2), 225–239 (2004)
23. Yang, Z., Tang, K., Yao, X.: Large scale evolutionary optimization using cooperative coevolution. Information Sciences 178, 2986–2999 (2008)

Hybrid Particle Swarm Optimisation Algorithms Based on Differential Evolution and Local Search

Wenlong Fu[1], Mark Johnston[1], and Mengjie Zhang[2]

[1] School of Mathematics, Statistics and Operations Research
[2] School of Engineering and Computer Science
Victoria University of Wellington, P.O. Box 600, Wellington, New Zealand

Abstract. Particle Swarm Optimisation (PSO) is an intelligent search method based on swarm intelligence and has been widely used in many fields. However it is also easily trapped in local optima. In this paper, we propose two hybrid PSO algorithms: one uses a Differential Evolution (DE) operator to replace the standard PSO method for updating a particle's position; and the other integrates both the DE operator and a simple local search. Seven benchmark multi-modal, high-dimensional functions are used to test the performance of the proposed methods. The results demonstrate that both algorithms perform well in quickly finding global solutions which other hybrid PSO algorithms are unable to find.

Keywords: Particle Swarm Optimisation, Differential Evolution.

1 Introduction

Particle Swarm Optimisation (PSO) is a stochastic global optimisation method which originated from the simulation of the social behaviour of birds within a flock, as developed by Kennedy and Eberhart in 1995 [1]. It is widely used in function optimisation [2], object detection [3], optimisation of wireless sensor networks [4], and many other applications [5,6].

The global optimisation of multi-modal functions is an important topic in scientific and engineering research since many real situations can be modelled as nonlinear optimisation problems. The standard PSO has difficulty with consistently converging to global optima, especially for multi-modal, high-dimensional functions. For escaping from local optima, Bratton and Blackwell [2] proposed a simplified recombinant PSO for function optimisation. Also, classical Differential Evolution (DE) operators have been integrated into hybrid PSO algorithms for global optimisation [8,9]. However, these PSO variants still have problems finding global solutions for some benchmark multi-modal, high-dimensional functions.

The goal of this paper is to investigate new hybrid PSO techniques for finding globally optimal solutions of multi-modal, high-dimensional functions. Instead of using the standard PSO method, we aim to use hybrid PSO techniques for updating a particle's position. We will consider two hybrid PSO approaches: using PSO with a DE operator (called HybridPSO1) rather than strongly depending on the currently global best and local best positions; and integrating

J. Li (Ed.): AI 2010, LNAI 6464, pp. 313–322, 2010.

the DE operator with a local search (called HybridPSO2) to do a small amount of additional searching for a better position about the current position. Both approaches will be examined and compared with some existing PSO methods on seven benchmark multi-modal, high-dimensional functions. We will focus on whether the new approaches *can* find the global solutions for these functions, and investigate the performance of these approaches in converging to a global solution.

The goal here is to determine whether hybrid PSO can find global optima which allude other PSO-based methods. This should give some idea as to whether it would subsequently be worthwhile applying the proposed hybrid PSO methods to other benchmark sets of test problems including read-world problems. If we are able to demonstrate effectiveness then a serious comparison with state-of-the-art algorithms would subsequently be needed.

In the remainder of this paper, Section 2 briefly describes background on PSO and DE, and Section 3 describes both hybrid algorithms in detail. After presenting the experimental design in Section 4, Section 5 discusses the experimental results. Finally, Section 6 gives conclusions and future work directions.

2 Background

This section briefly describes necessary background information on Particle Swarm Optimisation and Differential Evolution.

2.1 Particle Swarm Optimisation

Particle Swarm Optimisation (PSO) is a stochastic method for optimising without explicit knowledge of the gradient of the nonlinear function. PSO maintains a population of candidate solutions (called *particles*) and moves these particles around the search space. Each particle "flies" in a D-dimensional space according to the historical experiences of its own and its colleagues. Particle i has both a position, x_i, and a velocity v_i, which in "standard" PSO (SPSO), are updated as follows [10]:

$$v_{ik}^{t+1} = w \times v_{ik}^{t} + \phi_1 \times rand() \times (p_{ik}^{t} - x_{ik}^{t}) + \phi_2 \times rand() \times (g_{k}^{t} - x_{ik}^{t}) \quad (1)$$

$$x_{ik}^{t+1} = x_{ik}^{t} + v_{ik}^{t+1} \quad (2)$$

for component $k = 1, \ldots, D$. Here w is inertia weight; ϕ_1 and ϕ_2 are acceleration constants; $rand()$ are random values between 0 and 1; v_{ik}^{t} is the dimension k of the ith particle's velocity in generation t, v_i^t is the ith particle's velocity in generation t, and $v_i^t = [v_{i1}^t, v_{i2}^t, \ldots, v_{iD}^t]$; x_{ik}^t is the dimension k of the ith particle's position in generation t, x_i^t is the ith particle's position in generation t, and $x_i^t = [x_{i1}^t, x_{i2}^t, \ldots, x_{iD}^t]$; $p_i^t = [p_{i1}^t, p_{i2}^t, \ldots, p_{iD}^t]$ is the best position of the ith particle up to generation t, and $g^t = [g_1^t, g_2^t, \ldots, g_D^t]$ is the global best position of particles up to generation t. When termination criteria are satisfied, such as t being equal to the maximum generation, the global best position is taken as the solution to the problem.

2.2 Differential Evolution

Differential Evolution (DE) is also a population-based optimisation algorithm. It has been applied to classical optimisation and multi-objective optimisation [7]. DE creates new candidate solutions by combining existing ones, via three evolutionary operators: mutation, crossover and selection. The classical DE (crossover) operator is given as:

$$v_i^t = x_{l1}^t + F(x_{l2}^t - x_{l3}^t) \tag{3}$$

$$x_{i,j}^{t+1} = \begin{cases} v_{i,j}^t & rand() < p_{cr} \\ x_{i,j}^t & \text{otherwise} \end{cases} \tag{4}$$

where $x_{l1}^t, x_{l2}^t, x_{l3}^t$ represent the position of three individual particles (candidate solutions) from the population at the tth generation; $x_{i,j}^t$ is the jth element (dimension) value of the ith individual in the population at the tth generation; F is the so-called scaling factor ($F \in [0, 2]$), and p_{cr} is called the crossover probability ($p_{cr} \in [0, 1]$). DE is similar to PSO in that they both feature interaction among individuals.

2.3 Related Work of Hybrid PSO for Multi-modal Functions

In recent years, researchers have proposed hybrid PSO variants to optimise multi-modal functions. Zhang et al [9] and Xin et al [8] both combined PSO and DE operators to search for global solutions of multi-modal functions. Akbari and Ziarati [11] introduced stochastic local search in PSO for multi-modal function optimisation. In those methods, their purposes were to improve particles' exploration ability. Xin et al [8] used a probability to select standard PSO or DE operators to control particle movement and maintain population diversity in case all particles plunge into a local optima. However, those methods still have problems with some multi-modal functions, such as the Generalised Rastrigin function (see formula (6) in Section 4). Since the global best position affects all particles in PSO and it is easily trapped in one of the local optima of the Generalised Rastrigin function, all particles tend to prematurely converge near the current best position (not necessarily the true global optima). There is still an issue of how to improve particles' ability of exploring the search space for multi-modal function optimisation.

Since PSO has difficulty escaping from a locally optimal position in multi-modal function optimisation problems, Bratton and Blackwell [2] proposed a simple model, removing the effect of the global best position, that has better performance than standard PSO (SPSO). Therefore, one strategy for improving the searching ability of PSO is changing the way to update particle positions and weakening the effect of the global best position.

3 New Hybrid PSO Algorithms

Since DE has similarities with PSO, we propose to replace the position update method used in SPSO with a DE operator. For exploring a better position in the

Algorithm 1. HybridPSO1

1: Initialise the particles, the local best positions and the global best position.
2: Use the DE operator to update each particle's position using (3) and (4).
3: Update the current local best positions and the global best position.
4: If the maximum generation is reached, go to step 5; otherwise return to step 2.
5: Output the global best position particle as the solution.

neighbourhood of the current position, a local search operator is also introduced after a particle updates its position. The two hybrid algorithms are described in detail as follows.

3.1 Hybrid PSO Algorithm Based on Differential Evolution

The classical DE operator is introduced to update particle positions in this algorithm. The hybrid algorithm is called HybridPSO1. The particles positions are updated by (3) and (4), not by (1) and (2). In the algorithm HybridPSO1, we use three local best positions to construct new positions and then let each particle fly to the related new position if the new position is better than the current position. We save the global position in the memory and update its position if one new position is better than it. The global position will be returned as the global solution when the maximum generation is reached. The whole HybridPSO1 is described in Algorithm 1.

3.2 Hybrid PSO Algorithm Based on Differential Evolution and Local Search

In any PSO method, the behaviour of particles moving to the next position is discontinuous. It is possible that one particle cannot hit a better position in the current region. Based on HybridPSO1, we introduce a simple local search to better explore the neighbourhood of a local optima. The new hybrid PSO algorithm is called HybridPSO2. The local search is described in Algorithm 2, where x_i^k is the kth generation particle i. Particles are selected to update their position by the local search after arriving at a new position when the DE operation finishes. The local search is integrated in the hybrid PSO algorithm and the whole hybrid algorithm is described in Algorithm 3, where in step 3, only some particles arriving at new positions are selected to use the local search (with probability p_{local}).

3.3 Discussion

Both hybrid PSO techniques weaken the effect of the global best position on all particles. The way of using the DE operator here, in both hybrid PSO algorithms, is different from other hybrid PSO techniques based on DE [8,9]. Unlike these methods of alternating the use of standard PSO method and DE operators, both hybrid PSO techniques here directly replace the standard PSO method

Algorithm 2. Local Search

1: Let $\Delta x_i^0 = x_i^{k+1} - x_i^k$ and $itr = 0$.
2: If $f(x_i^{k+1}) > f(x_i^k)$, let $\Delta x_i^0 = -\Delta x_i^0$.
3: If $itr < n_{itr}$, go to step 4; otherwise go to step 7.
4: $x_i' = x_i^{k+1} + \Delta x_i^{itr}$.
5: If $f(x_i') > f(x_i^{k+1})$, let $\Delta x_i^{itr+1} = \Delta x_i^{itr}/2$, otherwise $x_i^{k+1} = x_i'$;
6: $itr = itr + 1$; go to step 3.
7: Finish the local search.

Algorithm 3. HybridPSO2

1: Initialise the particles, the local best positions and the global best position.
2: Use the DE operator to update each particle's position using (3) and (4).
3: Randomly select some particles and perform local search on them using Algorithm 2.
4: Update the current local best positions and the global best position.
5: If the maximum generation is reached, go to step 6; otherwise return to step 2.
6: Output the global best position particle as the solution.

by the DE operator. Weakening the influence from the global best position is similar to the simplified recombinant PSO [2], but the method for updating particle positions is different. In [2], each particle is affected by its history and its neighbour's history, however, each particle in HybridPSO1 and HybridPSO2 is affected by all particles.

4 Experimental Design

We now describe the test functions and parameter settings for our experiments.

4.1 Multi-modal and High-Dimensional Functions

In practical optimal design problems, objective functions often lead to multi-modal domains. Multi-modal, high-dimensional functions often contain many local minima and a single global optimum. As Section 2 mentioned, the Generalised Rastrigin function contains many local minima (Figure 1 shows the two-dimensional Rastrigin function, i.e., with $D = 2$). These local optima make many PSO variants *fail* to find the global solution [2,8,9,11].

A standard set of seven benchmark multi-modal functions are employed to show the global optimisation performance of the proposed HybridPSO1 and HybridPSO2. These problems each contain many local minima and a single global optimum.

1. *Generalised Schwefel 2.6*

$$f_1 = -\sum_{i=1}^{D} x_i \sin(\sqrt{|x_i|}) \quad x_i \in [-500, 500] \tag{5}$$

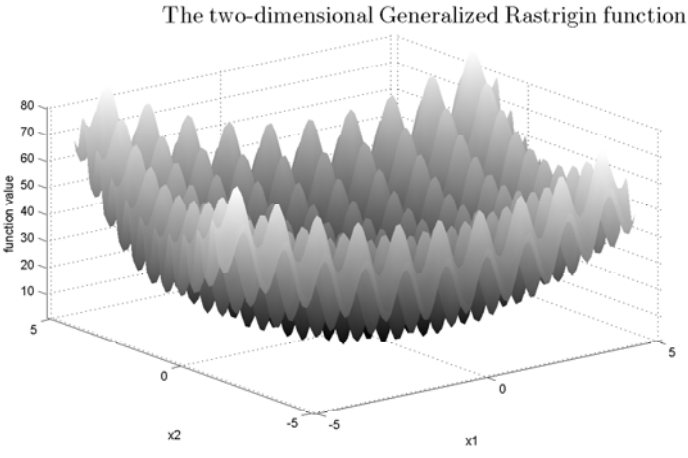


Fig. 1. The two-dimensional Generalized Rastrigin function

2. *Generalised Rastrigin*

$$f_2 = \sum_{i=1}^{D} \{x_i^2 - 10\cos(2\pi x_i) + 10\} \quad x_i \in [-5.12, 5.12] \tag{6}$$

3. *Ackley*

$$f_3 = -20\exp\left\{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right\} - \exp\left\{\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right\} + 20 + e \quad x_i \in [-32, 32] \tag{7}$$

4. *Generalised Griewank*

$$f_4 = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}) + 1 \quad x_i \in [-600, 600] \tag{8}$$

5. *Penalised function P8*

$$f_5 = \frac{\pi}{D}\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}\{1 + 10\sin^2(\pi y_{i+1}\} + (y_d - 1)^2\} + \sum_{i=1}^{D}\mu(x_i, 10, 100, 4)$$
$$\text{where} \quad y_i = 1 + \tfrac{1}{4}(x_i + 1) \quad x_i \in [-50, 50] \tag{9}$$

6. *Penalised function P16*

$$f_6 = 0.1\{10\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2\{1 + 10\sin^2(3\pi x_{i+1})\} +$$
$$(x_d - 1)^2\{1 + \sin^2(2\pi x_D)\}\} + \sum_{i=1}^{D}\mu(x_i, 5, 100, 4) \quad x_i \in [-50, 50] \tag{10}$$

where

$$\mu(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(-x_i - a)^m & x_i < a \end{cases}$$

7. *Shifted Rastrigin*

$$f_7 = \sum_{i=1}^{D} \{z_i^2 - 10\cos(2\pi z_i) + 10\} \quad x_i \in [-5.12, 5.12], z_i = x_i - o_i \tag{11}$$

where o_i is a random value in $[-5.12, 5.12]$.

The global optimal value of Generalized Schwefel 2.6 function is -12569.5, and the global optimal values of all of the other six functions are zero.

4.2 Parameter Setting

In the literature, $D = 30$ and $D = 100$ are usually chosen to test algorithm performance for solving multi-modal, high-dimensional functions, with $D = 30$ being the most popular setting. To evaluate the performance of the new hybrid algorithms and compare with results reported for DM3-PSO [2], DEPSO [8] and DE [8], $D = 30$ is selected. To compare with DM3-PSO [2], the number of function evaluations in HybridPSO1 and HybridPSO2 are limited to 300000. Table 1 shows the parameter values in the HybridPSO1 and HybridPSO2. We run each function test 100 times randomly and independently. These parameter values were chosen based on the literature.

5 Experimental Results and Discussion

We study the results from both hybrid algorithms and compare them with results reported for DM3-PSO [2], DEPSO [8] and DE [8]. Table 2 shows the comparison. The results for both SPSO and DM3-PSO come from [2]. Simply, values less than 10^{-11} have been round to 0.0. For DEPSO and DE we only list the results available from [8], which does not give the number of function evaluations.

Table 1. Parameter settings

Parameter	Value
Population size	60
D (dimension)	30
n_{itr}	4
p_{local}	0.05
F	1.2

Table 2. Results for solving seven multi-modal functions: mean ± standard deviation of best function values found from 100 replications. Here '±0.0' means that all replications found the global optima.

	HybridPSO1	HybridPSO2	SPSO [2]	DM3-PSO [2]	DEPSO [8]	DE [8]
f_1	-12569.5 ± 0.0	-12569.5 ± 0.0	3522 ± 32	1830 ± 46	-12569.5 ± 0.0	-9639.5 ± 190
f_2	0.0 ± 0.0	0.0 ± 0.0	140.156 ± 5.87	9.88 ± 0.86	0.0 ± 0.0	2.2 ± 1.8
f_3	0.0 ± 0.0	0.0 ± 0.0	12.93 ± 1.59	0.0 ± 0.0	0.0 ± 0.0	$(1.1 \pm 0.2)10^{-5}$
f_4	0.0 ± 0.0	0.0 ± 0.0	0.019 ± 0.004	0.0 ± 0.0	—	—
f_5	0.0 ± 0.0	0.0 ± 0.0	0.15 ± 0.05	0.0 ± 0.0	—	—
f_6	0.0 ± 0.0	0.0 ± 0.0	0.003 ± 0.001	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
f_7	1.6 ± 1.5	0.14 ± 0.40	—	—	55.0 ± 4.4	47.0 ± 7.2

From Table 2, it is found that DM3-PSO, HybridPSO1 and HybridPSO2 can successfully converge to the global solutions of functions f_3, f_4, f_5 and f_6, but DM3-PSO has difficulty with solving functions f_1 and f_2. HybridPSO1 and HybridPSO2 solve six of the multi-modal functions successfully, and gives a good solution to f_7. Especially for f_2 (Generalised Rastrigin function), these other PSO variants cannot even find the global solution [11]. In [8], DEPSO found the global optima in 7.7 seconds on a 2.8GHz CPU. In HybridPSO1, it only takes 1 second for 300 000 function evaluations with 2.1GHz CPU. HybridPSO1 finds the global value far less than 300 000 function evaluations (see From Figure 2). DEPSO [8] took 10.8 ± 0.2 seconds for solving function f_3 and HybridPSO1 only took about 1 second for finding the global optima. For finishing 300000 function evaluations, HybridPSO2 takes about 1 second. As a final test, the hybrid PSO methods perform considerably better on f_7 than DEPSO and DE. Based on the analysis and comparison, HybridPSO1 and HybridPSO2 have excellent performance in solving multi-modal and high-dimensional function optimisation problems.

Figure 2 plots the best value against each generation (top figure) and against the count of function evaluations (bottom figure), when using HybridPSO1 and HybridPSO2 to optimise function f_2. In both plots, the vertical axis represents the best value seen so far, averaging over 100 independent trails. Their evolution progress demonstrates HybridPSO2 converges faster to the global position in the early stages based on the population generation. HybridPSO2 has a slightly faster convergence speed to the global position in the early stages based on the number of function evaluations, but its speed is slower than HybridPSO1 after about 2.0×10^4 evaluations.

Since we weaken the global best position effect, the whole population does not appear to get trapped in one local optimal position, and both hybrid PSO algorithms keep good population diversity. The local search appears to help particles explore local optima, therefore HybridPSO2 accelerates the progress of particles evolution in the early stage but at the expense of more function evaluations.

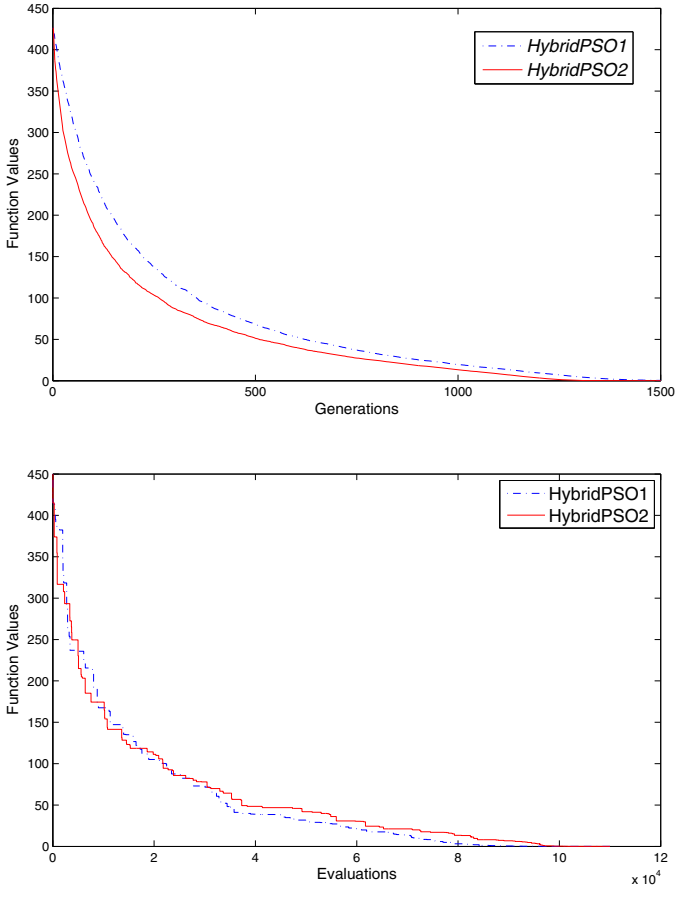


Fig. 2. The best function value for the 30-dimensional Generalised Rastrigin function in each generation (top figure) and each count of function evaluations (bottom figure) using HybridPSO1 and HybridPSO2

6 Conclusions

The goal of this paper was to investigate hybrid PSO approaches to optimise multi-modal functions. The goal was successfully achieved by using a DE operator and integrating a local search. In both hybrid algorithms, the convergence to local optima was successfully avoided, and the HybridPSO2 can converge faster to global solutions than the HybridPSO1 in the early stages.

Two hybrid PSO algorithms were developed in this paper. HybridPSO1 replaces the method in standard PSO with one DE operator and uses it to update particles. HybridPSO2 integrates one local search operator based on HybridPSO1, explores the local optimal position in particles region. Both hybrid PSO algorithms are effective to find the global solutions of the seven benchmark

multi-modal and high-dimensional functions. In future work, we will investigate further ways to use different local search operators to help particles fly to the global best position.

References

1. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks IV, pp. 1942–1948 (1995)
2. Bratton, D., Blackwell, T.: A simplified recombinant PSO. Journal of Artificial Evolution and Applications (2008)
3. Setayesh, M., Zhang, M., Johnston, M.: A new homogeneity-based approach to edge detection using PSO. In: Proceedings of the 24th International Conference on Image and Vision Computing New Zealand (IVCNZ), pp. 231–236. IEEE Press, Wellington (2009)
4. Aziz, N., Mohemmed, A.W., Zhang, M.: Particle swarm optimization for coverage maximization and energy conservation in wireless sensor networks. In: Di Chio, C., Brabazon, A., Di Caro, G.A., Ebner, M., Farooq, M., Fink, A., Grahl, J., Greenfield, G., Machado, P., O'Neill, M., Tarantino, E., Urquhart, N. (eds.) Applications of Evolutionary Computation. LNCS, vol. 6025, pp. 51–60. Springer, Heidelberg (2010)
5. Poli, R., Kennedy, J., Blackwell, T., Freitas, A.: Particle swarms: the second decade. Journal of Artificial Evolution and Applications (2008)
6. Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization: an overview. Swarm Intelligence 1(1), 33–57 (2007)
7. Price, K., Storn, R.M., Lampinen, J.A.: Differential Evolution: A Practical Approach to Global Optimization. Springer, Heidelberg (2005)
8. Xin, B., Chen, J., Peng, Z., Pan, F.: An adaptive hybrid optimizer based on particle swarm and differential evolution for global optimization. Science China Information Sciences 53(5), 980–989 (2010)
9. Zhang, W., Xie, X.: DEPSO: hybrid particle swarm with differential evolution operator. In: IEEE International Conference on Systems, Man & Cybernetics (SMCC), Washington DC, USA, pp. 3816–3821 (2003)
10. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of IEEE World Congress on Computational Intelligence, pp. 69–73 (1998)
11. Akbari, R., Ziarati, K.: Combination of particle swarm optimization and stochastic local search for multimodal function optimization. In: IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACAIIA), pp. 388–392 (2008)

Edge and Corner Extraction Using Particle Swarm Optimisation

Mahdi Setayesh[1], Mark Johnston[2], and Mengjie Zhang[1]

[1] School of Engineering and Computer Science
[2] School of Mathematics, Statistics and Operations Research
Victoria University of Wellington, P.O. Box 600, Wellington, New Zealand

Abstract. We apply particle swarm optimisation to the detection of edges and corners as low level features in noisy images and use these features to recognise simple objects. In this approach, the edges and the corners of an object are detected by a particle swarm optimisation algorithm and then the object is classified based on the number of corners and attributes of the edges by a simple fuzzy rule-based classifier. Several simple geometric objects in different locations, scales, and orientations have been used with a variety of impulse noise levels to assess the system. This system can categorise images containing these simple objects with high noise levels more accurately than an existing swarm-based edge and corner detector.

1 Introduction

Detection of low level features in images such as edges and corners is a critical element in image processing. These features can be detected without knowledge of the objects in the real world [1]. One of the most important challenges of edge and corner detectors is to detect the edges and corners in noisy images. Many edge and corner detectors have been proposed to overcome noise such as Gaussian-based [2], statistical-based [3], and scale space-based edge [4] and corner [5] detectors. Significant problems of the Gaussian-based edge detectors are displacement, removed edges, false edges [2] and also malfunctioning at corners [6]. The Gaussian filter used to remove noise in these methods often produces thick edges [7]. Statistical methods such as t-detector [1] and robust rank-order (RRO) detector [3] have been proposed to overcome the noise. These methods operate on a large area in comparison to other edge detection methods, but these methods are not based on an edge model, i.e., they are data-driven. Thus they cannot recognise edge direction which is required for non-maxima suppression (NMS) post-processing techniques. Scale space edge detectors operate on a large area through generating different scales of images. These methods are very fast, but their main problems are how to choose the size of the filters and how to combine edge information from different scales [2].

Particle swarm optimisation (PSO) is a population-based evolutionary algorithm for problem solving based on social-psychological principles, introduced by Kennedy and Eberhart in 1995 [8]. Some advantages of PSO in comparison

J. Li (Ed.): AI 2010, LNAI 6464, pp. 323–333, 2010.

to other population-based evolutionary algorithms such as genetic algorithms (GA) are ease of its implementation and high rate of convergence [9].

To date, PSO has been used in a number of vision applications such as object detection, object classification and segmentation [10], but there is very little work in which PSO has been applied to feature extraction such as edge and corner detection *directly*. This paper proposes two novel edge and corner detectors that use PSO to extract low level features of a noisy image such as its edges and corners and then a fuzzy rule-based classifier is utilised to detect the object through those features. This classifier uses the geometric attributes of the objects to recognise some simple objects.

The main goal in this paper is to detect the edges and corners in noisy images *without* using any pre-processing algorithm to remove the noise and any post processing algorithm to link the edges. Both newly designed detectors are compared to a swarm-based edge and corner detector proposed in [11] through applying the fuzzy rule-based classifier to the extracted features. The newly designed edge detector is based on identifying the boundaries of homogeneous regions in an image based on pixel intensity.

2 Background

This section describes some background on PSO, and edge and corner detection approaches.

2.1 Particle Swarm Optimisation

Particle Swarm Optimisation (PSO) is a global optimisation method, inspired by the social behavior of animals and other biological populations [12]; it simulates a simplified social model such as flocking of birds and schooling of fish. PSO was originally an optimisation method for continuous nonlinear functions; however, some discrete versions of PSO have also been proposed [13]. In PSO, there is a finite population of individual solutions (called *particles*), each having a memory of previous states [14]. Recently, PSO has been noted by researchers because of ease of its implementation, fewer operations in comparison to other heuristic algorithms, and high speed of global convergence [15,9].

In the basic PSO, there is a population of m particles that "fly" through an n-dimensional search space. The *position* of the ith particle is represented as the vector $\boldsymbol{X}_i = (x_{i1}, x_{i1}, ..., x_{in})$ and is changed according to its own experience and that of its neighbours. Let $\boldsymbol{X}_i(t)$ denote the position of particle P_i at time t. Then $\boldsymbol{X}_i$ is changed at each iteration of PSO by adding a *velocity* $\boldsymbol{V}_i(t)$, i.e.,

$$\boldsymbol{X}_i(t+1) = \boldsymbol{X}_i(t) + \boldsymbol{V}_i(t+1). \tag{1}$$

The velocity is updated based on three components: current motion, particle memory influence, and swarm influence, i.e.,

$$\boldsymbol{V}_i(t{+}1) = w\boldsymbol{V}_i(t){+}C_1 Rand_1(\boldsymbol{X}_{pbest_i}{-}\boldsymbol{X}_i(t)){+}C_2 Rand_2(\boldsymbol{X}_{leader}{-}\boldsymbol{X}_i(t)) \tag{2}$$

where $Rand_1$, and $Rand_2$ are uniform random variables between 0 and 1. Here, w denotes an inertia weight which controls the impact of the previous velocity; C_1 (called *self* confidence) and C_2 (called *swarm* confidence) are learning factors that represent the attraction of a particle toward either its own success or that of its neighbours; $\boldsymbol{X}_{pbest_i}$ denotes the best position of the ith particle so far; and $\boldsymbol{X}_{leader}$ is the position of a particle (the *leader*) which is used to guide other particles toward better regions of the search space. The leader of each particle is specified by a connected neighborhood topology represented as a graph [8].

2.2 Previous Approaches to Edge Detection

The *edge detection* task is to find the boundaries of image regions based on properties such as intensity and texture [3]. It is a critical low-level process of image processing because edges carry useful information. Applying an edge detector to an image considerably reduces the amount of the data to be processed, but it preserves the main shape of the objects present in an image. The shape of the edges depends on many parameters such as geometrical and optical properties of an image, illumination condition, and also noise level in the image [16]. Edge detection typically results in an edge map which describes the classification of each pixel of the image, as well as some other edge attributes such as magnitude and orientation [17].

A popular edge detection algorithm is based on the *homogeneity* operator which subtracts each of the eight surrounding pixel intensities from the center pixel of a 3×3 window, i.e.,

$$H_P = \begin{cases} \max\{|I_P - I_{Ni}|, i = 1, \ldots, 8\} & \text{if} > threshold \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where P is the particular pixel for which we are going to calculate H_P, N_i is i^{th} neighbourhood of pixel P, I_P is the intensity of pixel P, and $threshold$ is a user specified value.

2.3 Corner Detection as Interest Point Detection

In computer vision terminology, a corner is defined as an interest point that is created through an intersection of two or more edges. There are many approaches proposed in the literature [18,19,20]. Most of these approaches need complicated computations and some of them are poor in detection rate, localisation, repeatability rate, robustness to noise and speed aspects [21]. A recent overview can be seen in [22].

3 PSO for Low Level Feature Extraction

The approach proposed in this paper is to simplify an image into a list of geometric properties such as edges (with orientations) and corners (with angles), suitable for analysis by a classifier system, such as a rule-based system. Our proposed system consists of two main parts: a PSO-based edge detection algorithm and a PSO-based corner detection algorithm. Previously, we introduced a

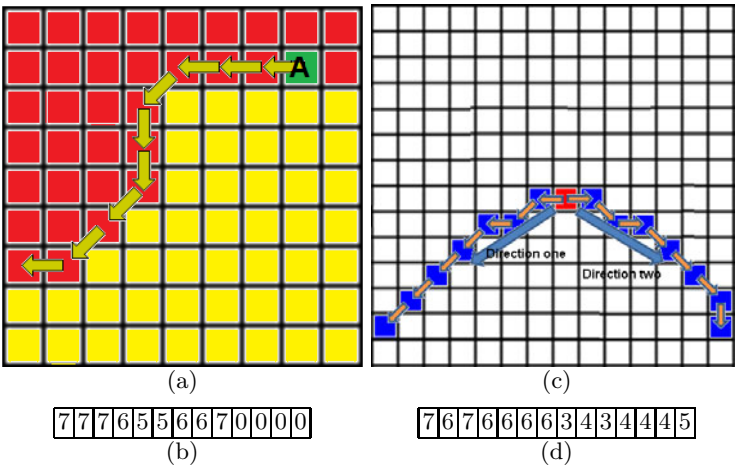


Fig. 1. (a) An example for a curve passing through pixel A and (b) particle encoding for this curve. (c) An example for a corner composed by two curves and (d) particle encoding for this corner.

PSO-based edge detection algorithm [23]. In this paper, we improve it to detect the edges of an image through a better fitness function. The PSO-based corner detection introduced in this paper is designed to accurately detect the corners of an image, especially in noisy environments.

3.1 PSO-Based Algorithm for Edge Detection

We adopt the edge detection algorithm from [23], which finds the best fitting edge as a collection of pixels which are on a continuous curve. This PSO-based algorithm is applied to each pixel in an image to detect the best curve which can be fitted that passes through each pixel. In the ideal case, the result of applying an edge detector to an image will lead to a set of connected curves that indicate the boundaries of objects.

In each particle, the value of each attribute is an integer in the range 0 to 8. These values represent the direction of the movement from a pixel to one of the eight possible adjacent pixels in its neighbourhood along a connected curve. Hence, a particle is represented as $\langle d_1, d_2, \ldots, d_{max} \rangle$, where max is the maximum number of pixels on a curve (depending on the image size). If the number of pixels on a curve is less than the dimension of a particle, the remaining cells will be set to zero. For example, the curve passing through a point corresponding with a pixel A in Figure 1a is encoded as in Figure 1b.

We expect that a large number of pixels on an edge of an image will have the same or similar intensity. However, the homogeneity of these pixels must be larger than the homogeneity of the pixels which are not on an edge. Therefore, we define two factors of a curve, the homogeneity and uniformity factors. The

Algorithm 1. PSO-based edge detection algorithm

```
For each pixel P on an image do
  If P is not marked as an edge then
    Initialize PSO population randomly for pixel P
    Repeat
      For each particle (decoded as curve C) do
        Evaluate H_C (4), L_C (5), U_C (6) and f_C (7)
        Update best particle if f_C is better than the best fitness value
      For each particle do
        Find local best particle from neighbourhood
        Calculate particle velocity (2) and apply velocity constriction
        Update particle position (1) and apply position constriction
    Until maximum iterations exceeded or minimum error criteria attained
    Select best particle in the population and decode it as curve C*
    If L_{C*} > Min_L then mark all pixels on curve C as an edge
```

homogeneity factor of a curve is the average of the homogeneity of the pixels on a curve, i.e.,

$$H_C = \frac{1}{L_C} \sum_{P_i \in C} H_{P_i} \tag{4}$$

where P_i is the i^{th} pixel on the curve C and L_C is the length of a curve C, given by

$$L_C = \sum_{P_i \in C} \begin{cases} 1 & \text{if } d_{P_i} \text{ is horizontal or vertical} \\ \sqrt{2} & \text{otherwise} \end{cases} \tag{5}$$

and H_{P_i} is calculated by using equation (3). The *uniformity factor of a curve* measures the similarity of pixel intensities along the curve, i.e.,

$$U_C = \frac{1}{L_C} \sum_{i=1}^{L_C - 1} |I_{P_{i+1}} - I_P| \tag{6}$$

As we search over curves which pass through a given pixel, we expect to maximize the homogeneity factor and length of the curve and minimize the uniformity factor. Hence we propose the following objective function (to maximize). However, if the curve C crosses itself, the value of f_C is set to $-\infty$.

$$f_C = \begin{cases} (H_C - U_C)L_C & \text{if } H_C \geq threshold \\ -\infty & \text{otherwise} \end{cases} \tag{7}$$

Algorithm 1 outlines our PSO-based edge detection algorithm. Here, Min_L is the minimum length of a curve in an image and has the effect of removing noise.

3.2 New PSO-Based Algorithm for Corner Detection

After applying the PSO-based edge detection algorithm (Algorithm 1), all present edges on an image have been marked. We propose to then apply the following PSO-based corner detection algorithm to find all corners which are created by two or more detected edges.

Algorithm 2. PSO-based corner detection algorithm

```
For each edge E detected by Algorithm 1 on an image do
  Initialize PSO population randomly for the edge E
  Repeat
    For each particle (decoded as corner) do
      Evaluate θ_corner and f_corner (8)
      Update best particle if f_corner is better than the best fitness value
    For each particle do
      Find local best particle from neighbourhood
      Calculate particle velocity (2) and apply velocity constriction
      Update particle position (1) and apply position constriction
  Until maximum iterations exceeded or minimum error criteria attained
  Select best particle in the population, decode and mark it as a corner
```

Each corner is composed of at least two curves that intersect each other at one point. Therefore, to describe a corner, each particle encodes two curves together. If the maximum length of each curve is 7, then the length of each corner particle will be 14. The value of each element is an integer in the range 0 to 8, again representing the direction of the movement on two curves which create the corner. Hence, each particle in the PSO population can be presented as $\langle d_1, \ldots, d_{14} \rangle$. For example, the corner in Figure 1c is composed of two curves in different directions (direction 1 and direction 2) that intersect each other at the red pixel. Elements 1 to 7 indicate the curve in direction 1 and elements 8 to 14 indicate the curve in direction 2. Therefore, the corner is encoded as in Figure 1d.

We propose an objective function (to be minimized) as

$$f_{corner} = \begin{cases} |\theta| & \text{if } 10 \leq |\theta| \leq 170 \\ -\infty & \text{otherwise} \end{cases} \tag{8}$$

where θ estimates the angle (in degrees) between two curves that make a corner, using an average direction several pixels along each of the two curves.

$$\theta = \tan^{-1}\left(\frac{gradient_1 - gradient_2}{1 + gradient_1 \times gradient_2}\right), \text{ where } gradient_i = \frac{\sum_{d_i \in C_{direction_i}} \Delta y_{d_i}}{\sum_{d_i \in C_{direction_i}} \Delta x_{d_i}}$$

$$\Delta x_{d_i} = \begin{cases} -1 & d_i = 0, 6, 7 \\ 0 & d_i = 1, 5 \\ 1 & d_i = 2, 3, 4 \end{cases} \quad \Delta y_{d_i} = \begin{cases} -1 & d_i = 0, 1, 2 \\ 0 & d_i = 3, 7 \\ 1 & d_i = 4, 5, 6 \end{cases}$$

Algorithm 2 outlines our PSO-based corner detection algorithm.

4 Experimental Design

To examine the effectiveness of the proposed PSO algorithms for edge and corner detection, we designed two sets of experiments. In the first set, we will directly check the detected edges and corners of images. Similarly to [11], in the second set, we will use a simple fuzzy rule-based classifier to detect simple objects

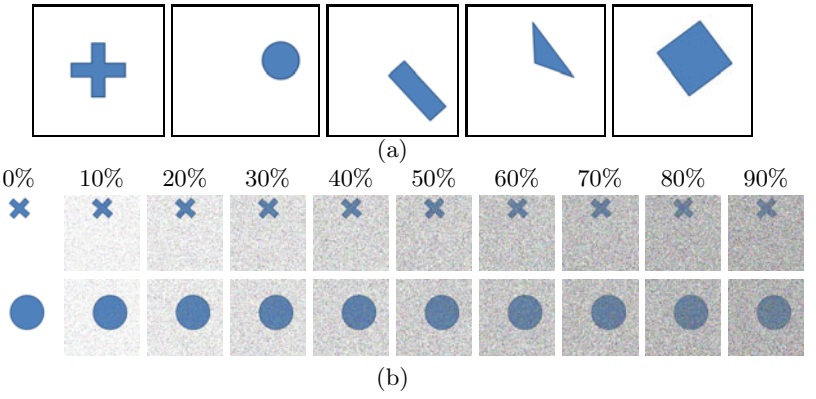


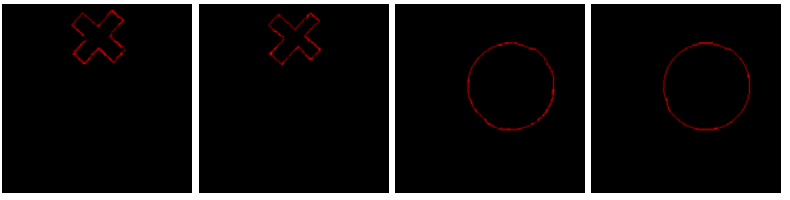
Fig. 2. (a) Simple test shapes (generated randomly in different positions, scales, and orientations) and (b) some of their noisy images in 10 different impulse noise levels which are used in the experiment to assess the system

in the images based on the detected corners and edges. The results will be compared with the approach proposed in [11]. The rest of this section describes the parameter settings of the PSO system, the image data set, and the simple fuzzy rules used for object detection.

PSO System. The PSO-based edge and corner detection algorithm used a population size of 10 and a maximum of 200 iterations. The minimum length of a curve in the PSO-based edge detection algorithm, Min_L was set at 5, and the maximum length of a curve restricted to 50. These values were chosen based on initial experiments. Following [8,12], we used the values $w = 1$, $C_1 = 2$ and $C_2 = 2$ for the parameters in (Eq. 2).

Test Image Sets. We tested the system on images containing five variants of simple shapes (squares, circles, crosses, rectangles and triangles) in different locations, scales and orientations, and in ten different impulse noise levels $(0, 10, 20, \ldots, 90\%)$. The image set includes 5000 images of size 256×256 pixels. Some of them are shown in Figure 2a. Examples of the noisy images containing circles and crosses are shown in Figure 2b.

Simple Fuzzy Rule-Based Classifier for Detection of Simple Objects. The fuzzy rule-based classifier has been used because of its simplicity. The system has been designed based on the Mamadani form of fuzzy inference system proposed in [24]. The system classifies the shapes based on extracted geometric attributes from the image. Extracted geometric attributes include the number of pixels on the horizontal, vertical and diagonal edges separately, and also the number of corners that are found in an image by the edge and corner detection algorithms shown in the previous subsections. The rules designed to classify the objects are shown in Table 1. This system applies these rules to the input parameters (information about of the corners and the edges) and classifies the simple objects. Fuzzy membership functions used in this system are triangular because they are easily designed and implemented.



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

Fig. 3. Four results of the PSO-based algorithm for edge detection: (a) the edges of a cross in an image with 10% noise; (b) the edges of a cross in an image with 30% noise; (c) the edges of a circle in an image with 10% noise; (d) the edges of a circle in an image with 30% noise.

5 Results and Discussion

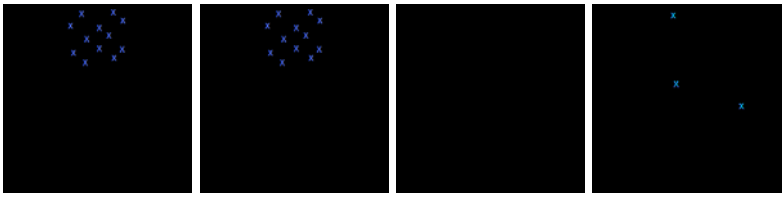
Results of edge and corner detection. Figure 3 shows the resulting images by applying the PSO-based edge detection algorithm on four images of the test image set at different noise levels. The results on other images show a similar pattern as long as the noise level is 50% or less. We observe that the algorithm appears to detect the edges well without any preprocessing even in noisy environments.

Figure 4 shows the resulting images from applying the PSO-based corner detection algorithm on four images. We observe that our algorithm can detect the corners well without any preprocessing even in noisy environments with a noise level of 50% or less.

Results on Object Detection. Table 2 shows classification rates for the detection of simple objects. The system appears to classify the objects well in different locations, scales and orientations without any noise. Representation of small circles in digital images makes it hard to recognise the circular objects for this system; however Table 2 shows that the classification rate for the circle is 0.98 in non-noisy images. The performance of the system is acceptable for images with noise level less than 30% but it reduces with increasing noise level. In comparison to the results from [11] (see Table 2), our PSO-based approaches appear to give results that are at least as good and often better (except for highly

Table 1. Fuzzy rule-based classifier. N_C: number of corners detected by the algorithm

Object	Condition
Cross	$N_C \approx 12$
Rectangle	$N_C \approx 4$ and for all corners $\theta \approx 90$
Square	Rectangle and number of edges in different directions almost equal or zero
Triangle	$N_C \approx 3$
Circle	$N_C \approx 0$ and number of edges in different directions almost equal



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

Fig. 4. Four results of the PSO-based algorithm for corner detection: (a) the corners of a cross in an image with 10% noise; (b) the corners of a cross in an image with 30% noise; (c) a circle without any corners in an image with 10% noise; (d) the corners of a triangle in an image with 10% noise.

Table 2. Classification rates for detection of simple objects in noisy environments (there are 100 images for each simple shape in every noise levels and the algorithm runs 20 times for each image). (b) Classification rates for object detection in noisy environments proposed in [11].

Noise	Rectangle		Cross		Triangle		Circle		Square	
	(a)	(b)	(a)	(b)	(a)	(b)	(a)	(b)	(a)	(b)
0%	1	1	1	1	1	1	0.98	0.7	1	1
10%	1	1	1	0.9	1	1	0.92	0.8	1	0.8
20%	1	1	1	0.8	1	1	0.9	0.7	0.92	0.5
30%	1	0.9	1	0.4	1	1	0.86	0.6	0.84	0.7
40%	0.90	0.9	0.81	0.7	1	0.9	0.77	0.5	0.72	0.3
50%	*0.75*	0.9	0.64	0.5	*0.81*	0.9	0.69	0.2	0.63	0.3
60%	*0.69*	0.7	0.50	0.2	*0.62*	0.9	0.71	0.4	0.52	0.2
70%	0.55	0.4	0.31	0.2	*0.49*	0.8	0.57	0.2	0.45	0.2
80%	0.47	0.3	0.17	0.3	*0.33*	0.6	0.46	0.1	0.39	0.3
90%	0.40	0.2	0	0	0.25	0.2	0.40	0.3	0.31	0.3

noisy triangles). It should be noted that our image set is harder than that used in [11]. Our images contain five variants of simple shapes in different locations, scales, and orientations, but images used in [11] contain the same shapes only in different locations. All triangles used in [11] are only equilateral, while in our system the triangles are equilateral, isosceles and scalene.

6 Conclusions

In this paper, two PSO-based approaches to the detection of edges and corners have been proposed. This was successfully achieved by introducing two new PSO-based algorithms for edge and corner detection, and the performance of the system was tested by recognition of the simple objects in different noise levels and compared with the system proposed in [11]. The results show that our system generally outperforms the system proposed in [11]. In addition, our

system can detect simple objects in noisy images without using any preprocessing and post processing algorithms. However, the current version of the system takes a relatively longer time than the swarm-based system proposed in [11]. We will further investigate new ways of overcoming this limitation in the future through finding better fitness functions and also applying a search mechanism to handle constraints better in the PSO. We will also compare accuracy of the newly developed algorithms to some state-of-the art edge and corner detectors such as t-detector, modified version of Canny edge detector (using ant colony optimisation), and also scale space-based edge and corner detectors.

References

1. Ruzon, M.A., Tomasi, C.: Edge, junction, and corner detection using color distributions. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 1281–1295 (2001)
2. Basu, M.: Gaussian-based edge-detection methods: A survey. IEEE Transactions on Systems, Man, Cybernetics Part C 32(3), 252–260 (2002)
3. Lim, D.H.: Robust edge detection in noisy images. Comput. Stat. Data Anal. 50(3), 803–812 (2006)
4. Tremblais, B., Augereau, B.: A fast multiscale edge detection algorithm based on a new edge preserving pde resolution scheme. In: International Conference on Pattern Recognition, vol. 2, pp. 811–814 (2004)
5. He, X.C., Yung, N.H.C.: Curvature scale space corner detector with adaptive threshold and dynamic region of support. In: Proceedings of the 17th International Conference on Pattern Recognition, vol. 2, pp. 791–794. IEEE, Los Alamitos (2004)
6. Sharifi, M., Fathy, M., Mahmoudi, M.T.: A classified and comparative study of edge detection algorithms. In: Proceedings of International Conference on Information Technology: Coding and Computing, pp. 117–120 (2002)
7. Ding, L., Goshtasby, A.: On the Canny edge detector. Pattern Recognition 34(3), 721–725 (2001)
8. Kennedy, F., Eberhart, R., Shi, Y.: Swarm Intelligence. Morgan Kaufmann, San Francisco (2001)
9. Al Rashidi, M., El-Hawary, M.: A survey of particle swarm optimization applications in electric power systems. IEEE Transactions on Evolutionary Computation 13(4), 913–918 (2009)
10. Cagnoni, S., Mordonini, M., et al.: Particle swarm optimization for object detection and segmentation. In: Giacobini, M. (ed.) EvoWorkshops 2007. LNCS, vol. 4448, pp. 241–250. Springer, Heidelberg (2007)
11. Mirzayans, T., Parimi, N., Pilarski, P., Backhouse, C., Wyard-Scott, L., Musilek, P.: A swarm-based system for object recognition. Neural Network World 15, 351–365 (2005)
12. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
13. Kashan, A.H., Karimi, B.: A discrete particle swarm optimization algorithm for scheduling parallel machines. Comput. Ind. Eng. 56(1), 216–223 (2009)
14. Li, Z., Liu, X., Duan, X.: Comparative research on particle swarm optimization and genetic algorithm. Computer and Information Science 3(2), 1122–1134 (2010)
15. Panda, S., Padhy, N.P.: Comparison of particle swarm optimization and genetic algorithm for facts-based controller design. Appl. Soft Comput. 8, 1418–1427 (2008)

16. Chidiac, H., Ziou, D.: Classification of image edges. In: Proceedings of the Conference on Vision Interface 1999, Canada, pp. 17–24 (1999)
17. Mlsna, P.A.: Gradient and Laplacian edge detection. In: Handbook of Image and Video Processing, 2nd edn., Elsevier Academic Press (2005)
18. Olague, G., Hernández, B.: A new accurate and flexible model-based multi-corner detector for measurement and recognition. Pattern Recognition Letters 26(1), 27–41 (2005)
19. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. International Journal on Computer Vision 60(1), 63–86 (2004)
20. Trajkovic, M., Hedley, M.: Fast corner detection. Image and Vision Computing 16(2), 75–87 (1997)
21. Zheng, Z., Wang, H., Teoh, E.: Analysis of gray level corner detection. Pattern Recognition Letters 20, 149–162 (1999)
22. Mokhtarian, F., Mohanna, F.: Performance evaluation of corner detectors using consistency and accuracy measures. Comput. Vis. Image Underst. 102(1), 81–94 (2006)
23. Setayesh, M., Zhang, M., Johnston, M.: A new homogeneity-based approach to edge detection using PSO. In: Proceeding of the 24th International Conference on Image and Vision Computing, New Zealand, pp. 231–236. IEEE Press, Los Alamitos (2009)
24. Sivanandam, S.N., Sumathi, S., Deepa, S.N.: Introduction to Fuzzy Logic using MATLAB. Springer, Heidelberg (2007)

Fuzzy Adaptive Artificial Fish Swarm Algorithm

Danial Yazdani[1], Adel Nadjaran Toosi[2,*], and Mohammad Reza Meybodi[3]

[1] Department of Electerical, Computer and Information Technology, IAU of Qazvin, Qazvin, Iran
[2] Department of Computer Science and Software Engineering, The University of Melbourne, Melbourne, VIC 3010, Australia
[3] Department of Computer Engineering and Information Technology, Amirkabir University of Technology, Tehran, Iran
d_yazdani@qiau.ac.ir, adeln@csse.unimelb.edu.au, mmeybodi@aut.ac.ir

Abstract. Artificial Fish Swarm Algorithm (AFSA) is a kind of swarm intelligence algorithms which is usually employed in optimization problems. There are many parameters to adjust in AFSA like *visual* and *step*. Through constant initializing of visual and step parameters, algorithm is only able to do local searching or global searching. In this paper, two new adaptive methods based on fuzzy systems are proposed to control the visual and step parameters during the AFSA execution in order to control the capability of global and local searching adaptively. First method uniformly adjusts the visual and step of all fish whereas in the second method, each artificial fish has its own fuzzy controller for adjusting its visual and step parameters. Evaluations of the proposed methods were performed on eight well known benchmark functions in comparison with standard AFSA and Particle Swarm Optimization (PSO). The overall results show that proposed algorithm can be effective surprisingly.

Keywords: Artificial Fish Swarm Algorithm (AFSA), particle Swarm Optimization (PSO), fuzzy system, global search, local search.

1 Introduction

Solving the optimization problems is one of the challenging issues that computer scientists always grapple with. Among the algorithms, swarm intelligence algorithms have been proved their capabilities in solving such problems. Among them Particle Swarm Optimization (PSO) [1] and Ant Colony Optimization (ACO) [2] are the most well-known algorithms that have been ever proposed. These algorithms have some characteristics that make them suitable for solving optimization problems, like scalability, fault tolerance, consistency, higher speed, flexibility, parallelism, etc.

Artificial fish swarm algorithm (AFSA) [3], proposed by Li Xiao Lei in 2002, is a stochastic population-based algorithm motivated by intelligent collective behavior of

[*] Adel Nadjaran Toosi was with the Department of Computer Software Engineering, IAU of Mashhad and is the corresponding author.

J. Li (Ed.): AI 2010, LNAI 6464, pp. 334–343, 2010.

fish groups in nature. AFSA has characteristics such as non-sensitive initial artificial fish location, flexibility and fault tolerant. It has been applied on different problems including fuzzy clustering [4], Resource Leveling [5], optimization of PID controller parameters [6], spread spectrum code estimation [7], data mining [8], optimization of DNA encoding sequence [9], etc.

In AFSA, a population of artificial fish exists which move towards an objective by performing some behaviors inspired form nature. These behaviors are Swarm, Follow, Prey and Movement. Artificial fish do the optimization process by performing the behaviors [4][6][7][9].

At its time, AFSA has not been comprehensively accepted by scientists for some reasons. High computational complexity, difficult implementation of the algorithm and the results not significantly better than similar algorithms can be noted here. In fact, algorithms such as PSO with less computational complexity are easier to implement and the results obtained from different versions of PSO show better performances than AFSA. It is worth mentioning that AFSA is not a version of PSO and differs significantly from PSO. One of the outstanding differences between these two algorithms is that particles in PSO move just based on the past movements and their previous experiences in the problem environment. Nevertheless, artificial fish movements depend on their current positions and other members of the group situations. Accordingly, movements of the fish differ from particles.

There are two important parameters in AFSA: Visual and Step. Artificial Fish search the problem environment as broad as their Visual, and then they move toward the target based on the random value of the step in each iteration. In standard AFSA initial values for these parameters have a great affect on the final result because of the fact that they remain constant and equal to the initial values until the end of algorithm. If larger initial values for Visual and Step is selected, artificial fish swarm move faster toward the global optimum and is more capable of passing the local optimums. Selecting lower values for these parameters causes better results in local searching.

In this paper, two fuzzy adaptive methods have been proposed that regulate the visual and step of artificial fish. First method generates a weight based on two inputs: ratio of the improved fish and iteration number. Visual and step are multiplied by the weight to attain the next iteration values for these parameter. It is important to mention that in the first method, which is called Fuzzy Uniform Fish (FUF) here, parameters of visual and step of all artificial fish are adjusted by a global weight attained from fuzzy controller output. The second method, Fuzzy Autonomous Fish (FAF), combines inputs like distance from the best artificial fish, fitness ranking of the current fish and iteration number for each fish. These parameters are used to create a weight value in order to adjust the visual and step in the next iteration.

Experiment results show the proposed method produced better outcomes in comparison to standard AFSA. To achieve better balance between local and global searching, the proposed algorithm has employed some fuzzy rules. As a consequence, the algorithm searching efficiency and convergence speed to the global optimum has been improved considerably.

Previously, a parameter which is called inertia weight has been applied on particle swarm optimization (PSO) algorithm by Shi and Eberhart [10]. Output of the proposed fuzzy engines here plays a similar role to the inertia weight in PSO.

This paper is organized as follows. Section 2 presents the fuzzy approaches in detail through the introduction of input parameters, output weight and fuzzy decision strategy. Section 3 experimentally compares the proposed methods with PSO algorithm and Standard AFSA using a set of benchmark functions. Finally, conclusions are drawn in Section 4.

2 Proposed Algorithms

In nature, fish swarms members have a certain visual which directly depends on the fish type, environment conditions (e.g. water fog) and around obstacles (e.g. water plants and other fish). When the swarm moves towards a target (e.g. food) as much as it converges on, visibility is reduced due to density. Here, the main motivation of our work is to implement this natural reality for the artificial fish.

In AFSA, artificial fish search the problem environment based on their visual and then they move towards the target by a random value of their step. Determination of the initial values of the step and visual influence on the final result essentially. Values of these parameters remain constant and equal to the initial values during the algorithm execution. If greater initial values have been considered for these parameters, artificial fish swarm will move faster towards the global optimum because artificial fish are able to search bigger environment around them and move with bigger step in each iteration. Under such circumstances, artificial fish are more powerful in escaping from the local optimums. Incidentally, there are some deficiencies in larger values of step and visual. Accuracy and consistency of the algorithm will decrease in such situation.

In fact, the algorithm acts better in global searching, but after approaching the global optimum, it is incapable of an appropriate local search because of the fact that the visual is larger than it should be. Therefore, owing to the large value of the visual, positions with better fitness are unlikely to be found and fish will pass the global optimum, even they may go far from it. Considering smaller values for these parameters make algorithm more consistent and accurate but it causes the algorithm to move towards the target more slowly and incapable of escaping local optimums.

Based on the above facts, in order to get better results, larger initial value for visual and step is selected first. Afterward, it is reduced during the algorithm execution adaptively. As a result, fish move towards the target quickly and are more capable of escaping local optimums. Finally, by approaching the target, artificial fish can accurately investigate the environment by smaller Visual and Step.

In order to control values of step and visual and balancing between global search and local search, a novel parameter, called *Constriction Weight* here, is proposed. Weight has to be greater than 0 and smaller than 1. Current iteration visual and step values are calculated according to the following formulas in presence of weight parameter:

$$Visual_{itr} = CW \times Visual_{itr-1} \tag{1}$$

$$Step_{itr} = CW \times Step_{itr-1} \tag{2}$$

Where *CW* is *Constriction Weight* which is generated as a output of the proposed fuzzy engines described in the following parts of this paper. $Visual_{itr}$ and $Step_{itr}$ stand for the current iteration visual and step and $Visual_{itr-1}$ and $Step_{itr-1}$ is the pervious iteration visual and step respectively.

With the purpose of attaining better values for the visual and step two different fuzzy methods for calculating the weight have been proposed here.

2.1 Fuzzy Uniform Fish (FUF)

In this method, weight is a value between 0 and 1 that is calculated as an output of the fuzzy engine. All of the fish in the swarm then adjust their visual and step based on the output weight. The proposed fuzzy engine has two inputs and one output: Iteration number and ratio of improved fish as inputs and constriction weight as an output. *Iteration Number*, normalized between 0 and 1, is the proportion of the current iteration number to the final iteration number. In fact, visual and step parameters must be larger in initial iterations to achieve better global searching. Therefore, visual and step values decreases smoothly in initial iterations of the algorithm execution. Progress of the algorithm causes the artificial fish come close to the global optimum of the problem. So, in order to increasing the local search capability of the algorithm, visual and step parameters must be reduced by larger amounts. As a result artificial fish are able to search the global optimum more keenly. Considering the above facts, by approaching to the final iterations, the proposed fuzzy engine increases the constriction weight to reduce the visual and step more sharply.

Ratio of Improved Fish is the proportion of the number of fish that find better positions in problem space (points with higher fitness) to the total number of fish in comparison with previous iteration.

When most of the artificial fish find better positions compared with previous iteration, visual and step parameter are suitable. Therefore, there is no need to reduce them. In this situation *Ratio of Improved Fish* value is a number close to 1. Conversely, when most of the artificial fish do not experience any improvement rather than previous iteration the constriction weight must be increased to reduce the visual and step. Act of reducing visual and step raises the probability of the finding positions with better fitness values due to the fact that it increases the local searching capability of the fish. Considering the above facts, the proposed fuzzy engine increases the constriction weight by reduction of the ratio of the improved fish value and vice versa. Figures 1(a) and 1(b) show the membership functions for Inputs: *Iteration Number* and *Ratio of Improved Fish*.

Constriction Weight is the output of the fuzzy engine which has the membership functions of figure 1(c). The proposed fuzzy engine, which is a Mamdani fuzzy inference system with centroid of area defuzzification strategy, uses the rules shown in the fuzzy associative memory in Table 1.

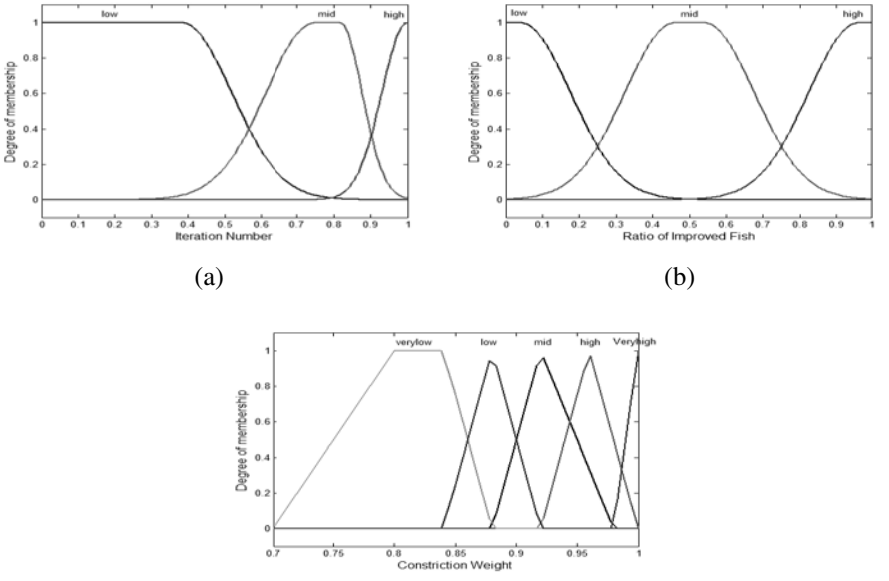


Fig. 1. Fuzzy Uniform Fish Membership functions. (a) Iteration Number. (b) Ratio of Improved Fish. (c) Constriction Weight.

Table 1. Fuzzy associative memory for the proposed FUF engine. VL: very low, L: low, M: mid, H: high and VH: very high.

Iteration Number	Ratio of Improved Fish	Constriction Weight
L	H	VH
L	M	H
L	L	M
M	H	H
M	M	M
M	L	L
H	H	M
H	M	L
H	L	VL

2.2 Fuzzy Autonomous Fish (FAF)

The main difference between the Fuzzy Autonomous Fish (FAF) method and Fuzzy Uniform Autonomous (FUF) is that each artificial fish adjust its visual and step parameter individually and independent of the rest fish in FAF.

In this method, initial values of the visual and step parameters for all the fish is identical. But through the algorithm execution, for each artificial fish, these parameters decrease based on the output of the fuzzy engine which is obtained according to the position of the fish in the group. Consequently, each fish has its own visual and step parameter independent and different from the others. The fuzzy

inference system based on three inputs calculates the Constriction Weight. These inputs are *Distance from Best*, *Fitness Ranking* and *Iteration Number*.

Distance from Best is a normalized rank-based value between 0 and 1 for each fish. It is calculated based on Euclidian distance of the artificial fish from the best artificial fish (fish with best fitness value). Afterward, all fish are sorted and ranked based on the Euclidian Distance from the best fish. Proportion of the ranking number to a total number of artificial fish is considered as the *Distance from Best* input parameter of the fuzzy engine.

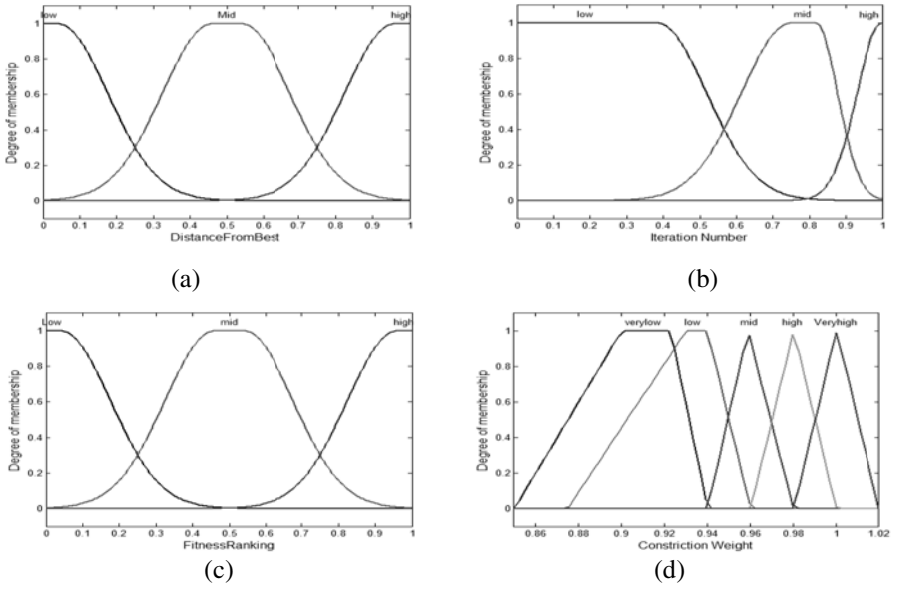


Fig. 2. Inputs and Outputs of the Fuzzy Autonomous Fish inference system membership functions. (a)Distance from Best. (b) Iteration Number. (c) Fitness Ranking. (d) Constriction Weight.

In this situation, as long as the distance of the fish from the best fish is large, parameter of visual and step must be reduced lesser. If a fish is far from the best fish then the amount of visual and step parameters should be reduced more sharply in order to perform the search more accurately. This policy leads to fish cannot step out from the area with high level of appropriateness with long step.

Fitness Ranking is equal to the proportion of the ranking number, calculated based on the fitness value for the artificial fish, to the total number of artificial fish. In order to calculate this parameter, all artificial fish are sorted based on their fitness value and then ranked based on the index of their position on the list. Similar to *Distance from Best,* the worse ranking that artificial fish have, lesser amount visual and step parameters should be reduced and vice versa. Iteration number has the same definition of the FUF engine. Figure 2 depicts the membership functions of FAF engine. The FAF engine is a Mamdani fuzzy inference system with centroid of area defuzzification strategy. FAF uses 27 rules have been shown in the fuzzy associative memory in Table 2.

Table 2. Fuzzy associative memory for the proposed FAF engine. VL: very low, L: low, M: mid, H: high and VH: very high.

Distance from Best	Fitness Ranking	Iteration Number	Weight	Distance from Best	Fitness Ranking	Iteration Number	Weight
L	L	H	VL	M	M	L	H
L	L	M	VL	M	H	H	M
L	L	L	L	M	H	M	H
L	M	H	L	M	H	L	VH
L	M	M	M	H	L	H	L
L	M	L	H	H	L	M	M
L	H	H	L	H	L	L	H
L	H	M	M	H	M	H	M
L	H	L	VL	H	M	M	H
M	L	H	VL	H	M	L	VH
M	L	M	L	H	H	H	M
M	L	L	M	H	H	M	H
M	M	H	L	H	H	L	VL
M	M	M	M				

3 Experimental Results

In this section, eight benchmark functions are tested for the sake of comparison. All of them are standard test functions. Table 3 shows name of the functions, their equations and search space for each function [11].

Table 3. Eight test functions used in this paper

Name of Function	Function	Search Space
Ackly	$f_1(x) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)}$	$-32 < x_i \le 32$
Sphere	$f_2(x) = \sum_{i=1}^{D} x_i^2$	$-100 \le x_i \le 100$
Griewank	$f_3(x) = \sum_{i=1}^{n}\left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-600 < x_i \le 600$
Schwefel's2.22	$f_4(x) = \sum_{i=1}^{D}\lvert x_i\rvert + \prod_{i=1}^{D}\lvert x_i\rvert$	$-10 < x_i \le 10$
Step	$f_5(x) = \sum_{i=1}^{D}\left(\lfloor x_i + 0.5\rfloor\right)^2$	$-100 < x_i \le 100$
Rastrigin	$f_6(x) = \sum_{i=1}^{D}\lvert x_i\rvert + \prod_{i=1}^{D}\lvert x_i\rvert$	$-5.12 < x_i \le 5.12$
Non continuous Rastrigin	$f_7(x) = \sum_{i=1}^{D}\left(y_i^2 - 10\cos(2\pi y_i) + 10\right)$ $Where \begin{cases} x_i & \lvert x_i\rvert < 0.5 \\ \frac{round(2x_i)}{2} & \lvert x_i\rvert \ge 0.5 \end{cases}$	$-5.12 < x_i \le 5.12$
Generalized Penalized	$f_8(x) = \frac{\pi}{D}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_D - 1)^2\right\}$ $+ \sum_{i=1}^{D} u(x_i, 10, 100, 4)$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$-50 < x_i \le 50$

In experiments, initial values for Visual and Step has been considered 40% and 25% of the range length of the fitness function variables respectively (For example range length of the fitness function variables in Ackly function is 64). Crowd factor is 0.5, maximum try-number is 10 and population number in standard AFSA, FAF and FUF is 30. In global version of PSO (GPSO) [10], inertia weight value linearly decreases from 0.9 to 0.4 during the algorithm execution. The population size is equal to 5*D where D is the problem dimensions. At last, parameters c_1 and c_2 have been set in form of $c_1 = c_2 = 2$. Experiments repeated 100 times; best, mean and standard deviation obtained from running of standard AFSA, FUF, FAF, and GPSO in 30-dimensional spaces on eight benchmark functions for 1000 Iterations have been reported on Tables 4.

As it is shown in table 4 and figure 3, in standard-AFSA, since the visual and step parameter are constant during the algorithm execution, algorithm is not able to reach the acceptable results in none of the benchmark functions. The main reason for this deficiency is that the algorithm is not sufficient flexible in different situations with which group are faced in problem the space. In other words, algorithm shows a uniform behavior in every possible situation and is not able to keep the balance between global and local searching.

Table 4. Comparison of Standard AFSA, GPSO, FAF and FUF on eight benchmark functions

Function	Algorithm	Min	Mean	Std.Dev
Ackly	Std.AFSA	5.7284	6.8145	0.5602
	GPSO	1.8082e-04	7.4578e-04	3.6421e-04
	FUF	1.3322e-14	1.6875e-14	3.2601e-15
	FAF	1.3322e-14	1.7941e-14	3.3704e-15
Sphere	Std.AFSA	4.1020+02	7.2819+02	1.8326e+02
	GPSO	3.3124e-07	5.1620e-06	9.0477e-06
	FUF	4.4040e-49	7.0516e-49	1.5620e-49
	FAF	1.2361e-68	8.8204e-63	2.7849e-62
Griewank	Std.AFSA	4.0010	6.9563	1.5048
	GPSO	1.2028e-05	0.0390	0.1008
	FUF	2.2204e-16	3.6637e-16	1.0865e-16
	FAF	4.4408e-16	7.54951e-16	1.8724e-16
Schwefel's2.22	Std.AFSA	7.2255	9.7531	1.0017
	GPSO	1.1192e-04	3.0191	4.1696
	FUF	7.1387e-21	9.5280e-16	3.9264e-15
	FAF	1.8153e-08	3.9340e-04	0.0010
Step	Std.AFSA	383	667.6500	184.6787
	GPSO	0	0	0
	FUF	0	0	0
	FAF	0	0	0
Rastrigin	Std.AFSA	77.5419	162.0464	34.2185
	GPSO	21.9819	41.6566	9.5776
	FUF	0	0.5472	1.0447
	FAF	0	0.5969	1.3846
Noncontinuous Rastrigin	Std.AFSA	76.3466	139.1457	18.8128
	GPSO	28.3333	74.3343	32.0512
	FUF	0	1.0500	1.1909
	FAF	0	2.3773	2.6676
Generalized Penalized	Std.AFSA	2.6573	5.1480	1.2510
	GPSO	1.9245e-06	0.0618	0.1120
	FUF	3.6357e-32	1.8239e-30	4.7910e-30
	FAF	3.7648e-32	1.3889e-17	4.7272e-17

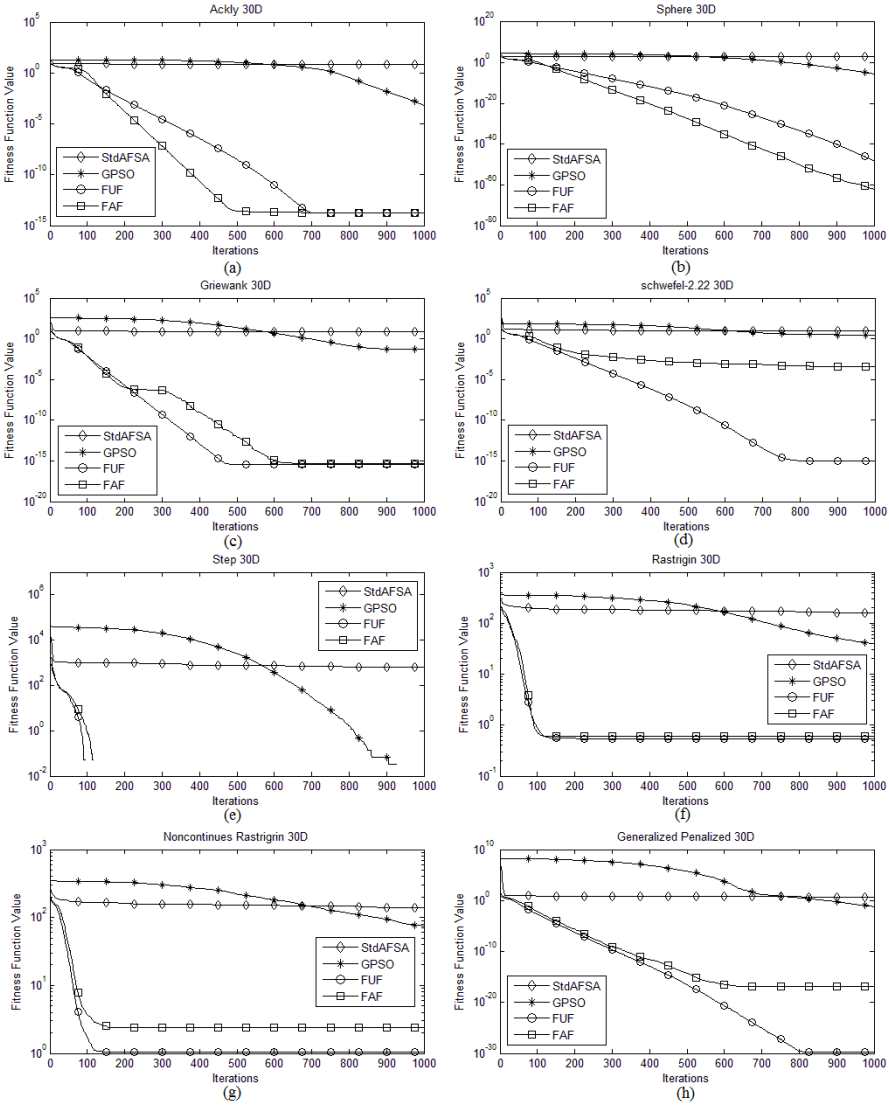


Fig. 3. Comparison of the average results of Std-AFSA, GPSO, FAF and FUF on 30-Dimensional (a) Ackly, (b) Sphere, (c) Griewank, (d) Schwefel 2.22, (e) Step, (f) Rastrigin, (g) Noncontinues Rastrigin and (g) Generalized Penalized functions in 1000 iterations.

Results show that proposed algorithms, FAF and FUF, considerably improved the effectiveness. The main reason is that the proposed algorithms dynamically decrease the visual and step parameter based on the output of the fuzzy systems which act according the group position. Therefore, during the algorithm execution, capability of the algorithm in global searching gradually decreases and capability of it in local searching increases. Artificial fish pass the local optimum more rapidly, and then they search around global optimum more accurately after converging to the global optimum.

Both of the proposed algorithms attain similar results on Ackly, Griewank, Step and Rastrigin. However, in Sphere which a function without a local optimum, FAF produces better results than FUF. Actually, because of the diversity in visual and step values among the artificial fish, FAF is more capable in local searching. This can be adverse in some functions like Generalized Penalized and Schwefel's2.22. In FAF, when an artificial fish continually experiences higher ranks, it reduce its visual and step faster, subsequently, it become frozen finally. In general we can say that the proposed algorithms are more efficient than Standard-AFSA and Global version of PSO.

4 Conclusion

In this paper, two modified AFSA algorithms have been proposed. Proposed algorithms take the members group position into account in adjusting the visual and step parameter. Our experimental results show AFSA with an appropriate adaptive visual and step can acts better than standard AFSA and other similar optimization algorithms like PSO. Two introduced fuzzy engine bring a significant improvement on the AFSA performance. Various simulations have been performed to support it.

References

1. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: IEEE International Conference on Neural Network, Perth, pp. 1942–1948 (1995)
2. Dorigo, M., Birattari, M., Stutzle, T.: Ant Colony Optimization. IEEE Computational Intelligent Magazine 1, 28–39 (2006)
3. Li, L.X., Shao, Z.J., Qian, J.X.: An Optimizing Method Based on Autonomous Animate: Fish Swarm Algorithm. In: Proceeding of System Engineering Theory and Practice, vol. 11, pp. 32–38 (2002)
4. Hi, S., Belacel, N., Hamam, H., Bouslimani, Y.: Fuzzy Clustering with Improved Artificial Fish Swarm Algorithm. In: International Joint Conference on Computational Sciences and Optimization 2009, Hainan, vol. 2, pp. 317–321 (2009)
5. Tian, W., Tian, Y.: An Improved Artificial Fish Swarm Algorithm for Resource Leveling. In: International Conference on Management and Service Science, Wuhan, pp. 1–4 (2009)
6. Luo, Y., Zhang, J., Li, X.: The Optimization of PID Controller Parameters Based on Artificial Fish Swarm Algorithm. In: IEEE International Conference on Automation and Logistics, Jinan, pp. 1058–1062 (2007)
7. Jiang, M., Wang, Y., Rubio, F., Yuan, D.: Spread Spectrum Code Estimation by Artificial Fish Swarm Algorithm. In: IEEE International Symposium on Intelligent Signal Processing, Alcala de Henares, pp. 1–6 (2007)
8. Zhang, M., Shao, C., Li, M., Sun, J.: Mining Classification Rule with Artificial Fish Swarm. In: 6th World Congress on Intelligent Control and Automation, Dalian, vol. 2, pp. 5877–5881 (2006)
9. Cui, G., Cao, X., Zhou, J., Wang, Y.: The Optimization of DNA Encoding Sequences Based on Improved Artificial Fish Swarm Algorithm. In: IEEE International Conference on Automation and Logistics, Jinan, pp. 1141–1144 (2007)
10. Shi, Y., Eberhart, R.: A Modified Particle Swarm Optimization. In: IEEE International Conference on Evolutionary Computation Proceedings, Anchorage, pp. 69–73 (1998)
11. Zhan, Z.H., Zhang, J., Li, Y., Chung, H.S.H.: Adaptive Particle Swarm Optimization. IEEE Transaction on System, Man and Cybernetics, Part B: Cybernetics 39(6), 1362–1381 (2009)

The Honourable Negotiator: When the Relationship Is What Matters

John Debenham[1] and Simeon Simoff[2]

[1] Centre for Quantum Computation & Intelligent Systems
University of Technology, Sydney, NSW, Australia
debenham@it.uts.edu.au
[2] School of Computing and Mathematics,
University of Western Sydney, NSW, Australia
s.simoff@uws.edu.au

Abstract. Relationships are fundamental to all but the most impersonal forms of interaction in business. Human agents who are unsure of themselves seek honourable trading relationships. The establishment and growth of interpersonal relationships is a result of reciprocal exchange of information. This paper addresses the problem of use of information for developing and utilising relationships between negotiating agents. The presence of measurable information on the Internet underpins the philosophy of transparency in electronic business, which has an impact on the behaviour of involved agents. It takes 'two to tango' for conducting business in such a 'net landscape' where communities and cliques emerge, shape and evolve. The paper presents a formalism for electronic negotiation technology that treats relationships as a commodity. It supports relationship building, maintaining, evolving, and passing to other agents, and utilises such relationships in agent interaction. The Honourable Negotiator also takes in account information about the relationships in networks of respective agents outside the trading space.

1 Introduction

Relationships are fundamental to all but the most impersonal forms of interaction in business. Human agents who are unsure of themselves seek *honourable* trading relationships.Through interactions, including information sharing over Internet social networking sites, agents are connected in various social networks, which provide data for modeling and analysis of the emerging relationships. The presence of such measurable information underpins the philosophy of transparency in electronic business, which has an impact on the behaviour of involved agents. The interplay between information, related directly to negotiation issues and negotiation in electronic business has been studied in our earlier work on the 'Curious Negotiator' [15]. It is founded on the intuition that "it's what you know that matters". Its development deployed (i) information theory, including entropy-based (random worlds) inference, for forming negotiation strategies [5,13]; (ii) information mining and delivery techniques [1,18]; and (iii) virtual institutions as the operational environment [5]. This body of work did not consider the use of information for developing relationships between negotiators. This is the research problem addressed in this paper.

J. Li (Ed.): AI 2010, LNAI 6464, pp. 344–353, 2010.

According to social penetration theory [3] the establishment and growth of interpersonal relationships is a result of reciprocal exchange of information. Such exchange starts with relatively non-intimate topics and gradually progresses to more personal and private topics. These encapsulate the philosophy behind the *'Honourable Negotiator'* — an agency where agents are capable of establishing and reshaping relationships with other agents to compensate for the lack of understanding of their own preferences over negotiation outcomes. In brief, an agent aims to secure projected needs by attempting to build a set of (business) relationships with other agents. A relationship is built by exchanging private information, and is characterised by its intimacy — degree of closeness — and balance — degree of fairness. Each argumentative interaction between two agents then has two goals: to satisfy some immediate need, and to do so in a way that develops the relationship in a desired direction. As a very simple example, if I don't know anything about how red wine should be cellared then I look for an honourable wine merchant who does.

There is evidence from psychological studies that humans seek a *balance* in their working relationships. The classical view [2] is that people perceive resource allocations as being distributively fair (i.e. well balanced) if they are proportional to inputs or contributions (i.e. equitable). However, more recent studies [16,17] show that humans follow a richer set of norms of distributive justice depending on their *intimacy* level: equity, equality, and need. The perception of a relation being in balance (i.e. fair) depends strongly on the nature of the social relationships between individuals (i.e. the intimacy level). In purely economical relationships (e.g., business), equity is perceived as more fair; in relations where joint action or fostering of social relationships are the goal (e.g. friends), equality is perceived as more fair; and in situations where personal development or personal welfare are the goal (e.g. family), allocations are usually based on need. According to [16,17], the more intimacy across the illocutionary categories the more the need norm is used, and the less intimacy the more the equity norm is used.

The contribution of this paper is in the development of the formalisms that (i) enable agents to build, manage and utilise in negotiation social relationships based of information exchange between them, and; (ii) enable agents to obtain the external information related to relationships that they need. Further the paper is organised as follows. Relationships are formalised and incorprated in agents strategies in Section 2, including the agent architecture in Section 2.1. Section 2.2 describes an elaborate means of measuring the intimacy — degree of closeness — and balance — degree of fairness — that is based on measures of the information in any utterance. Section 2.3 describes the argumentation framework. Section 3 presents the machinery for extracting and incorporating relevant information from external sources, including partner prediction and the software architecture. Section 4 concludes the paper.

2 Information-Based Agency That Handles Relationships

This work continues the development of *information-based agency* [13]. An information-based agent has an identity, values, needs, plans and strategies all of which are expressed using a fixed ontology in probabilistic logic for internal representation and in an illocutionary language for communication. All of the forgoing is represented

in the agent's deliberative machinery. We assume that such an agent resides in a virtual institution [4] and is aware of the prevailing norms and interaction protocols. An information-based agent makes no *a priori* assumptions about the states of the world or the other agents in it — these are represented in a world model, $\mathcal{M}^t$, that is inferred solely from the messages that it receives. The intuition behind information-based agency is that all illocutionary acts (like *Offer*, *Accept* and *Reject*) give away (valuable) information. Previous work has been centred on the design of interaction strategies and on proposing agent architectures able to deal with the exchange of offers [9,6]. Game theory [12], possibilistic logic [7] and first-order logic [11] have been used for this purpose. Some initial steps in proposing rhetoric particles have been made, especially around the idea of *appeals*, *rewards* and *threats* [14]. Expanded dialogues, including these and other rhetoric moves, are known as *argumentation-based interactions*. *Argumentation* in this sense is mainly to do with building (business) *relationships*. When we reward or threaten we refer to a future instant of time where the reward or threat will be effective, its scope goes beyond the current negotiation round.

A multiagent system $\{\alpha, \beta_1, \ldots, \beta_n, \xi, \theta_1, \ldots, \theta_t\}$, contains an agent α that interacts with other *argumentation agents*, β_i, *information providing agents*, θ_j, and an *institutional agent*, ξ, that represents the institution where we assume the interactions happen [4]. The institutional agent reports promptly and honestly on what actually occurs after an agent signs a contract, or makes some other form of commitment. Agents have a probabilistic first-order *internal language* $\mathcal{L}$ used to represent a *world model*, $\mathcal{M}^t$. An agent's world model, $\mathcal{M}^t$, is a set of probability distributions for a set of random variables each of which represents the agent's expectations about some point of interest about the world or the other agents in it. Each incoming utterance is translated into a set of (linear) constraints on one or more of these distributions, and then the posterior state of the world model is estimated using entropy-based inference. These distributions are the foundation for the agent's reasoning.

2.1 An Architecture to Enable Relationships

The agent architecture is shown in Figure 1. Agent α acts in response to a *need* that is expressed in terms of the ontology. Needs trigger α's goal/plan proactive reasoning, while other messages are dealt with by α's reactive reasoning.[1] Each plan prepares for the negotiation by assembling the contents of a 'briefcase' that the agent 'carries' into the negotiation[2]. The *relationship strategy* determines which agent to negotiate with for a given need; it uses risk management analysis to preserve a strategic set of trading relationships for each mission-critical need — this is not detailed here. For each trading relationship this strategy generates a *relationship target* that is expressed in the dialogical framework as a desired level of *intimacy* to be achieved in the long term.

[1] Each of α's plans and reactions contain constructors for an initial *world model* $\mathcal{M}^t$. $\mathcal{M}^t$ is then maintained from percepts received using *update functions* that transform percepts into constraints on $\mathcal{M}^t$ — for details, see [13].

[2] Empirical evidence shows that in human negotiation, better outcomes are achieved by skewing the opening offer in favour of the proposer. We are unaware of any empirical investigation of this hypothesis for autonomous agents in real trading scenarios.

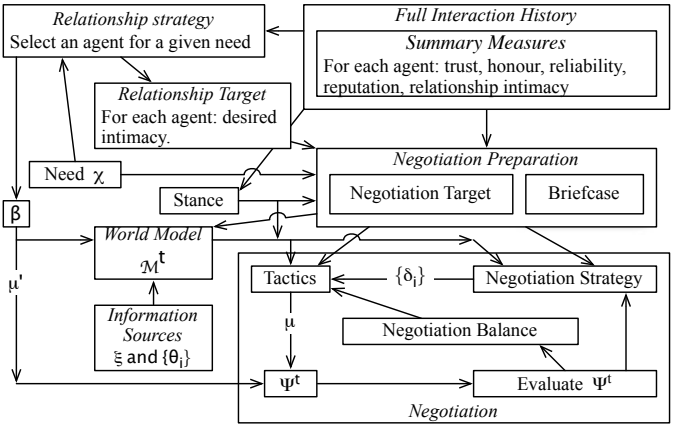


Fig. 1. The agent architecture

Each interaction consists of a dialogue, Ψ^t, between two agents with agent α contributing utterance μ and the partner β contributing μ'. Each dialogue, Ψ^t, is evaluated using the dialogical framework in terms of the *value* of Ψ^t to both α and β — see Section 2.2. The *interaction strategy* then determines the current set of offers $\{\delta_i\}$, and then the *tactics*, guided by the *interaction target*, decide which, if any, of these offers to put forward and wraps them in argumentation dialogue — see Section 2.3. We now describe two of the distributions in $\mathcal{M}^t$ that support offer exchange.

2.2 Valuing Dialogues

Suppose that an interaction commences at time s, and by time t a string of utterances, $\Phi^t = \langle \mu_1, \ldots, \mu_n \rangle$ has been exchanged between agent α and agent β. This dialogue is evaluated by α in the context of α's world model at time s, $\mathcal{M}^s$, and the environment e that includes utterances that may have been received from other agents in the system including the information sources $\{\theta_i\}$. Let $\Psi^t = (\Phi^t, \mathcal{M}^s, e)$, then α estimates the *value* of this dialogue to itself in the context of $\mathcal{M}^s$ and e as a $2 \times L$ array $V_\alpha(\Psi^t)$ where:

$$V_x(\Psi^t) = \begin{pmatrix} I_x^{l_1}(\Psi^t) & \ldots & I_x^{l_L}(\Psi^t) \\ U_x^{l_1}(\Psi^t) & \ldots & U_x^{l_L}(\Psi^t) \end{pmatrix}$$

where the $I(\cdot)$ and $U(\cdot)$ functions are information-based and utility-based measures respectively. α estimates the *value* of this dialogue to β as $V_\beta(\Psi^t)$ by assuming that β's reasoning apparatus mirrors its own.

The *balance* in a dialogue, Ψ^t, is defined as: $B_{\alpha\beta}(\Psi^t) = V_\alpha(\Psi^t) \ominus V_\beta(\Psi^t)$ for an element-by-element difference operator $\ominus$ that respects the structure of $V(\Psi^t)$. The *intimacy* between agents α and β, $I^{*t}_{\alpha\beta}$, is the pattern of the two $2 \times L$ arrays V^{*t}_α and V^{*t}_β that are computed by an update function as each interaction round terminates, $I^{*t}_{\alpha\beta} = \left(V^{*t}_\alpha, V^{*t}_\beta\right)$. If Ψ^t terminates at time t:

$$V_x^{*t+1} = \nu \times V_x(\Psi^t) + (1 - \nu) \times V_x^{*t} \tag{1}$$

where ν is the learning rate, and $x = \alpha, \beta$. Additionally, V_x^{*t} continually decays by: $V_x^{*t+1} = \tau \times V_x^{*t} + (1-\tau) \times D_x$, where $x = \alpha, \beta$; τ is the decay rate, and D_x is a $2 \times L$ array being the decay limit distribution for the value to agent x of the intimacy of the relationship in the absence of any interaction. D_x is the *reputation* of agent x. The *relationship balance* between agents α and β is: $B_{\alpha\beta}^{*t} = V_\alpha^{*t} \ominus V_\beta^{*t}$. The notion of balance may be applied to pairs of utterances by treating them as degenerate dialogues. In simple multi-issue bargaining the *equitable information revelation* strategy generalises the tit-for-tat strategy in single-issue bargaining, and extends to a tit-for-tat argumentation strategy by applying the same principle across the dialogical framework.

2.3 Relationship Strategies and Tactics

Each dialogue has to achieve two goals. First it may be intended to achieve some contractual outcome. Second it will aim to contribute to the growth, or decline, of the relationship intimacy. We now describe in greater detail the contents of the "*Negotiation*" box in Figure 1. The negotiation literature consistently advises that an agent's behaviour should not be predictable even in close, intimate relationships. The required variation of behaviour is normally described as varying the negotiation *stance* that informally varies from "friendly guy" to "tough guy". The stance is shown in Figure 1, it injects bounded random noise into the process, where the bound tightens as intimacy increases. The stance, $S_{\alpha\beta}^t$, is a $2 \times L$ matrix of randomly chosen multipliers, each ≈ 1, that perturbs α's actions. The value in the (x, y) position in the matrix, where $x = I, U$ and $y \in \mathcal{L}$, is chosen at random from $[\frac{1}{l(I_{\alpha\beta}^{*t}, x, y)}, l(I_{\alpha\beta}^{*t}, x, y)]$ where $l(I_{\alpha\beta}^{*t}, x, y)$ is the bound, and $I_{\alpha\beta}^{*t}$ is the intimacy.

The negotiation *strategy* is concerned with maintaining a working set of proposals. If the set of proposals is empty then α will quit the negotiation. α perturbs the acceptance machinery (see Section 2.1) by deriving s from the $S_{\alpha\beta}^t$ matrix. In line with the comment in Footnote 2, in the early stages of the negotiation α may decide to inflate her opening offer. The following strategy uses the machinery described in Section 2.1. Fix h, g, s and c, set the Proposals to the empty set, $\text{acc}(\alpha, \beta, \delta)$ means "contract δ with agent β is acceptable to α", and $\mathbb{P}$ denotes probability. Let $D_s^t = \{\delta \mid \mathbb{P}^t(\text{acc}(\alpha, \beta, \delta)) > c\}$, then repeat the following as many times as desired: add $\delta = \arg\max_x \{\mathbb{P}^t(\text{acc}(\beta, \alpha, x)) \mid x \in D_s^t\}$ to Proposals, remove $\{y \in D_s^t \mid \text{Sim}(y, \delta) < k\}$ for some k from D_s^t. By using $\mathbb{P}^t(\text{acc}(\beta, \alpha, \delta))$ this strategy reacts to β's history of Propose and Reject utterances.

Negotiation *tactics* are concerned with selecting some offers and wrapping them in argumentation. Prior interactions with agent β will have produced an intimacy pattern expressed in the form of $\left(V_\alpha^{*t}, V_\beta^{*t}\right)$. Suppose that the relationship target is $(T_\alpha^{*t}, T_\beta^{*t})$. Following from Equation 1, α will want to achieve a *negotiation target*, $N_\beta(\Psi^t)$ such that: $\nu \cdot N_\beta(\Psi^t) + (1-\nu) \cdot V_\beta^{*t}$ is "a bit on the T_β^{*t} side of" V_β^{*t}:

$$N_\beta(\Psi^t) = \frac{\nu - \kappa}{\nu} V_\beta^{*t} \oplus \frac{\kappa}{\nu} T_\beta^{*t} \tag{2}$$

for small $\kappa \in [0, \nu]$ that represents α's desired *rate of development* for her relationship with β, and $\oplus$ is element-by-element matrix addition. $N_\beta(\Psi^t)$ is a $2 \times L$ matrix containing variations in the dialogical framework's dimensions that α would like to reveal to β

during Ψ^t (e.g. I'll pass a bit more information on options than usual, I'll be stronger in concessions on options, etc.). It is reasonable to expect β to progress towards her target at the same rate and $N_\alpha(\Psi^t)$ is calculated by replacing β by α in Equation 2. $N_\alpha(\Psi^t)$ is what α hopes to receive from β during Ψ^t. This gives a *negotiation balance target* of: $N_\alpha(\Psi^t) \ominus N_\beta(\Psi^t)$ that can be used as the foundation for reactive tactics by striving to maintain this balance across the dialogical framework. A cautious tactic could use the balance to bound the response μ to each utterance μ' from β by the constraint: $V_\alpha(\mu') \ominus V_\beta(\mu) \approx S^t_{\alpha\beta} \otimes (N_\alpha(\Psi^t) \ominus N_\beta(\Psi^t))$, where $\otimes$ is element-by-element matrix multiplication, and $S^t_{\alpha\beta}$ is the stance in Fig. 1. A less neurotic tactic could attempt to achieve the target negotiation balance over the anticipated complete dialogue. If a balance bound requires negative information revelation in one dialogical framework category then α will contribute nothing to it, and will leave this to the natural decay to the reputation D as described above.

The following are a list of components that we have described that could be combined into an agent's negotiation strategy. These components all constrain the agent's actions. We assume that they are all soft constraints and that they operate together with a hard constraint $C^t(\alpha, \beta, x^t)$ on the message x^t that α may send to β at time t.

Information-based strategies. Every communication gives away information and so has the potential to contribute to the intimacy and balance of a relationship. Information-based strategies manage the information revelation process. Let $M^t_{\alpha\beta}$ be the set of time-stamped messages that α has sent to β, and $M^t_{\beta\alpha}$ likewise both at time t. $\mathcal{M}^t$ is α's world model at time t and consists of a set of probability distributions. x^t denotes a message received at time t. $\mathbb{I}^t(\alpha, \beta, x^t)$ is the information gain — measured as the reduction of the entropy of $\mathcal{M}^t$ — observed by α after receiving message x^t. $\mathbb{I}^t(\beta, \alpha, x^t)$ is α's estimate of β's information gain after receiving message x^t from α.

The complete *information history* of both the observed and the estimated information gain, $G^t(\alpha, \beta)$, is:

$$G^t(\alpha, \beta) = \{(x^s, \mathbb{I}^s(\alpha, \beta, x^s)) \mid x^s \in M^t_{\beta\alpha}\} \cup \{(x^s, \mathbb{I}^s(\beta, \alpha, x^s)) \mid x^s \in M^t_{\alpha\beta}\}$$

respectively.

In [13] we described the model that α constructs of β. In general α can not be expected to guess β's world model, $\mathcal{M}^t_\beta$, unless α knows what β's needs are — even then, α would only know $\mathcal{M}^t_\beta$ with certainty if it knew what plans β had chosen. However, α always knows the private information that it has sent to β — for example, in $\mathrm{Propose}(\cdot)$ and $\mathrm{Reject}(\cdot)$ messages. Such private information could be used by β to estimate α's probability of accepting a proposal: $\mathbb{P}^t_\beta(\mathrm{acc}(\alpha, \beta, \chi', z))$, where χ' is the need that β believes α to have.

α's information-based strategies constrain its actions, x^t, on the basis of $\mathbb{I}^t(\beta, \alpha, x^t)$ and its relation to $G^t(\alpha, \beta)$. For example, the strategy that gives β greatest expected information gain:

$$\arg\max_z \{\, \mathbb{I}^s_\beta(\beta, \alpha, z) \mid C^t(\alpha, \beta, z)\}$$

More generally, for some function f:

$$\arg\max_z\{\ f(\mathbb{I}^s_\beta(\beta,\alpha,z), G^t(\alpha,\beta)) \mid C^t(\alpha,\beta,z)\}$$

the idea being that the f 'optimises' in some sense the information gain taking account of the interaction history.

Ontology-based strategies. The structure of the ontology may be used to manage the information revelation process in particular strategic areas. For example, α may prefer to build a relationship with β in the context of the supply of particular goods only [10]. The structure of the ontology is provided by the $\mathrm{Sim}(\cdot)$ function. Given two contracts δ and δ' containing concepts $\{o_1,\ldots,o_i\}$ and $\{o'_1,\ldots,o'_j\}$ respectively, the (non-symmetric) distance of δ' from δ is the vector

$$\boldsymbol{\Gamma}(\delta,\delta') = (d_k : o''_k)_{k=1}^i$$

where $d_k = \min_x\{\mathrm{Sim}(o_k,o'_x) \mid x = 1,\ldots,j\}$, $o''_k = \sup(\arg\min_x\{\mathrm{Sim}(o_k,x) \mid x = o'_1,\ldots,o'_j\}, o_k)$ and the function $\sup(\cdot,\cdot)$ is the supremum of two concepts in the ontology. $\boldsymbol{\Gamma}(\delta,\delta')$ quantifies how different δ' is to δ and enables α to "work around" or "move away from" a contract under consideration. In general for some function g;

$$\arg\max_z\{\ g(\boldsymbol{\Gamma}(z,x^s)) \mid x^s \in M^t_{\alpha\beta} \cup M^t_{\beta\alpha} \wedge C^t(\alpha,\beta,z)\}$$

the idea being that the g 'optimises' in some sense the ontological relationship with the interaction history.

3 Providing Agents with Information from External Sources

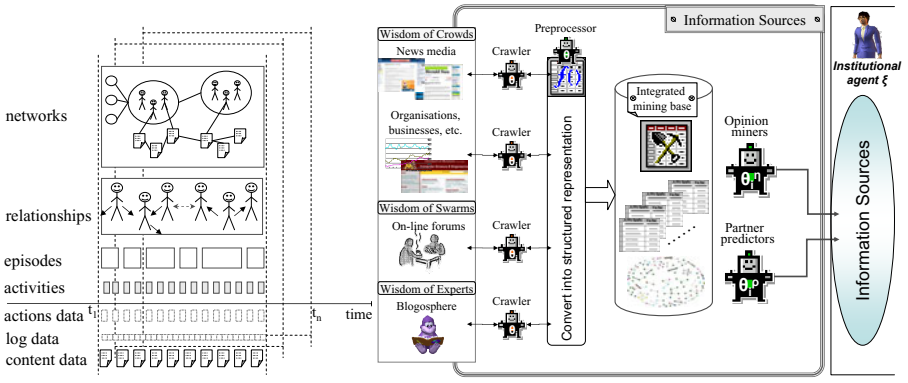
Information sources $\{\theta_i\} = \{\theta^\eta_i\} \bigcup \{\theta^\rho_i\}$ include two types: $\{\theta^\eta_i\}$ which provide opinion on negotiation issues and $\{\theta^\rho_i\}$ which provide opinion about relationships that may affect α. The $\{\theta^\eta_i\}$ agents have been designed to operate with several information spheres on the Web, categorised according to [8]. The 'Wisdom of Crowds' (online news sites, sites of organisations, businesses) provides information about a specific topic of interest; an example of how $\{\theta^\eta_i\}$ utilises online news sites for developing an opinion (advice) for α on exchange rate for a certain point in time is presented in [18]. The 'Wisdom of Swarms' (self-organised online forums, where individuals and institutions exchange information), contain focused and up-to-date information about a certain subject; an example of how $\{\theta^\eta_i\}$ utilises product (digital camera) forums for extracting information and forming an opinion from comments about the products, in a form suitable for α, is presented in [1]. Combining the 'Wisdom of Experts' (blogs, which contain the private views and opinion of an individual) about a certain topic is wise; an example of how $\{\theta^\eta_i\}$ provides α with expert opinion about a product preference (a range of wines from a specific wine region in Spain) formed from mining the respective part of the blogosphere is presented in [5]. Further we describe two aspects of our supporting architecture — the extraction of relevant relationship information and the prediction of potential partners.

Extracting relevant relationship information. Each of the above mentioned information spheres contains also information about various relations — the extraction and delivery of such "relationship gossip" is a prerogative of $\{\theta_i^\rho\}$. α may be interested in the quality of the information that β provides during their interactions, or may be looking for honourable 'well-positioned' trading partners, where 'well-positioned' is defined in terms of the positions that β has in the $\mathcal{N}_\beta^s$ networks in which it is involved in time s outside the marketplace. The position of a negotiating agent β in a network is given by vector $\boldsymbol{\pi}(\beta) = \langle \pi_i(\beta), i = 1, \ldots, n_j \rangle$, where its elements $\{\pi_i(\beta)\}$ are the n_j network features of interest to α. Therefore $\{\theta_i^\rho\}$ extracts the $\mathcal{N}_\beta^s \times n_j$ array Π^s of all $\mathcal{N}_\beta^s$ positions $\{\boldsymbol{\pi}(\beta)_i\}$ of β in the networks in which it is involved.

The collection of features may vary depending on the way the network is extracted and the information extracting tasks. Technically the problem is to convert the interaction data array $\mathcal{D}_m^{\mathcal{A}}$, describing the activities of m individuals into an $m \times m$ incidence matrix describing the graph $G_j(N_j, L_j)$ with N_j nodes and L_j links, and $\pi_i(\beta) = f_i(N_j, L_j)$, where $f_i(N_j, L_j)$ is a respective network feature, which is a function of some node or/and link parameters.

Fig.2a is a high level layered view of the extraction of relationships and networks from different pieces of data on the Web. For instance, directed network structures between agents can be extracted from the threaded logs of on-line forums, using the network extraction mechanisms in the Deep Email Miner. Such networks are further refined based on text mining of the content of the posts. Here, each agent β_i is represented through an n_ω–dimensional vector $\boldsymbol{\tau}_i^\beta = \langle \tau(\omega_j) | j = 1, \ldots, n_\omega \rangle$, where $\tau(\omega_j)$ is some function of the term count ω_j. The relations then are extracted depending on the similarity $Sim(\cdot)$ between agents in the term space. The information on directed edges in the resultant graph includes also $\{r_j\}$ — the estimates of the strength of the relationships. $Sim(\cdot)$ for each link is a simple initial estimate of an r_j — higher values can be interpreted as β_i having a strong relation with β_k, when lower values of that are still above the threshold of link existence can be interpreted as weak relations. These estimates are utilised by $\{\theta_i^\rho\}$ to provide α with an estimate $Cap(\beta_i)$ of the potential of a new partner β_i to sustain relationship.

Partner prediction. Section 1 presented a simple example when α may need to find a new partner, relying on external information only. The task for $\{\theta_i^\rho\}$ translates into a partner recommendation problem. It can be treated as *partner prediction* and estimation of *relationship capability* of predicted partner. For partner prediction, we need to find a classifier $\mathcal{C}_\alpha^s$ that classifies all dyads $\{\alpha; \beta_i\}$ in α's network in which α and β_i are not linked in time s into two classes $\{Partner, Alone\}$, where $Partner$ labels vertices with potential to link and $Alone$ labels vertices with no such potential. Considering α and potential set of agents $\{\beta_i\}$ that do not have direct link to α, predictors include the following normalised attributes: number of relations in which an agent is involved; shortest path between α and each agent β_i; a suit of network centrality measures of respective agents, including closeness (how well related an agent is to all other agents in considered network) and betweenness (to what extent an agent can reach respective agents through its relationships); interconnectedness for each β_i, measured by the normalised number of direct links to other agents of the kind; relationship importance — a function of the weights of existing relationships w_j; and the proportion of common



a. Mining relations and networks b. What is in the box 'Information Sources' in Fig.1

Fig. 2. Process and architecture for extracting and delivering information from external sources

partners for α and β_i out of all partners that they have. *Relationship capability* for a predicted partner is estimated from the above as: $\text{Cap}(\beta_i) = \sum_{j=1}^{n_\beta} w_j r_j / n_\beta \sum_{j=1}^{n_\beta} w_j$.

Information mining architecture. Fig.2b presents what goes into the the 'Information Sources' block in Fig.1. On request either from α or the institutional agent ξ crawlers retrieve a snapshot over certain period of time. The integrated mining base is updated with the retrieved data, after preprocessing the later into a structured and semi-structured forms. This database is the source for the above described relationship predicting. The two icons labeled 'Opinion miners' and 'Partner predictors' represent the θ_i^η and θ_i^ρ types of information mining agents. The information providing system uses the virtual institution protocols for information exchange between the agents [4].

4 Conclusions

We have presented the formalism which extends the information-based agency and is supporting information mining services to handle relationships and information related to relationships. The negotiation strategy and tactics treat information as a commodity and a relationship with an agent as all the information about the interaction with this agent. The formalism behind the Honourable Negotiator supports requesting and accommodating in the agent's reasoning relationship information from external sources. The practical value of presented formalisms is the automation of electronic business in contemporary 'net landscape' where communities and cliques emerge, shape and evolve. The Honourable Negotiator implements those formalisms in a virtual institution.

Acknowledgements

This research is supported by an ARC Discovery Grant DP0879789, University of Technology, Sydney and University of Western Sydney.

References

1. Aciar, S., Zhang, D., Simoff, S.J., Debenham, J.K.: Informed recommender: Basing recommendations on consumer product reviews. IEEE Intelligent Systems, 39–47 (May/June 2007)
2. Adams, J.S.: Inequity in social exchange. In: Berkowitz, L. (ed.) Advances in Experimental Social Psychology, vol. 2. Academic Press, New York (1965)
3. Berscheid, E., Reis, H.: Attraction and close relationships. In: The Handbook of Social Psychology, pp. 193–281. McGraw-Hill, New York (1998)
4. Bogdanovych, A.: Virtual Institutions. Ph.D. thesis, Faculty of IT, University of Technology, Sydney (November 2007)
5. Debenham, J., Simoff, S.: Believable electronic trading environments on the web. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence WI 2009, Milan, Italy, September 15-18, pp. 631–638. IEEE/WIC/ACM, IEEE (2009)
6. Faratin, P., Sierra, C., Jennings, N.: Using similarity criteria to make issue trade-offs in automated negotiation. Journal of Artificial Intelligence 142(2), 205–237 (2003)
7. Giménez, E., Godo, L., Rodríguez-Aguilar, J.A., Garcia, P.: Designing bidding strategies for trading agents in electronic auctions. In: Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS 1998), pp. 136–143 (1998)
8. Gloor, P.A., Krauss, J., Nann, S., Fischbach, K., Schoder, D.: Web science 2.0: Identifying trends through semantic social network analysis. In: Proceedings of the International Conference on Computational Science and Engineering, pp. 215–222. IEEE Computer Society, Los Alamitos (2009)
9. Jennings, N., Faratin, P., Lomuscio, A., Parsons, S., Sierra, C., Wooldridge, M.: Automated negotiation: Prospects, methods and challenges. International Journal of Group Decision and Negotiation 10(2), 199–215 (2001)
10. Kalfoglou, Y., Schorlemmer, M.: IF-Map: An ontology-mapping method based on information-flow theory. In: Spaccapietra, S., March, S., Aberer, K. (eds.) Journal on Data Semantics I. LNCS, vol. 2800, pp. 98–127. Springer, Heidelberg (2003)
11. Kraus, S.: Negotiation and cooperation in multi-agent environments. Artificial Intelligence 94(1-2), 79–97 (1997)
12. Rosenschein, J.S., Zlotkin, G.: Rules of Encounter. The MIT Press, Cambridge (1994)
13. Sierra, C., Debenham, J.: Information-based agency. In: Huang, T.S., Nijholt, A., Pantic, M., Pentland, A. (eds.) ICMI/IJCAI Workshops 2007. LNCS (LNAI), vol. 4451, Springer, Heidelberg (2007)
14. Sierra, C., Jennings, N., Noriega, P., Parsons, S.: A Framework for Argumentation-Based Negotiation. In: Rao, A., Singh, M.P., Wooldridge, M.J. (eds.) ATAL 1997. LNCS, vol. 1365, pp. 177–192. Springer, Heidelberg (1997)
15. Simoff, S., Debenham, J.: Curious negotiator. In: Klusch, M., Ossowski, S., Shehory, O. (eds.) CIA 2002. LNCS (LNAI), vol. 2446, pp. 104–111. Springer, Heidelberg (2002)
16. Sondak, H., Neale, M.A., Pinkley, R.: The negotiated allocations of benefits and burdens: The impact of outcome valence, contribution, and relationship. Organizational Behaviour and Human Decision Processes (3), 249–260 (1995)
17. Valley, K.L., Neale, M.A., Mannix, E.A.: Friends, lovers, colleagues, strangers: The effects of relationships on the process and outcome of negotiations. In: Bies, R., Lewicki, R., Sheppard, B. (eds.) Research in Negotiation in Organizations, vol. 5, pp. 65–94. JAI Press, Greenwich (1995)
18. Zhang, D., Simoff, S.J., Debenham, J.K.: Exchange rate modelling for e-negotiators using text mining techniques. In: Lu, J., Ruan, D., Zhang, G. (eds.) E-Service Intelligence - Methodologies, Technologies and Applications, pp. 191–211. Springer, Heidelberg (2007)

Tuning Java to Run Interactive Multiagent Simulations over Jason*

Víctor Fernández-Bauset, Francisco Grimaldo,
Miguel Lozano, and Juan M. Orduña

Computer Science Department, University of Valencia,
Dr. Moliner 50, (Burjassot) Valencia, Spain
ferbau@alumni.uv.es, {francisco.grimaldo,miguel.lozano,
juan.orduna}@uv.es

Abstract. Java-based simulation environments are currently used by many multiagent systems (MAS), since they mainly provide portability as well as an interesting reduction of the development cost. However, this kind of MAS are rarely considered when developing interactive applications with time response constraints. This paper analyses the performance provided by Jason, a well-known Java-based MAS platform, as a suitable framework for developing interactive multiagent simulations. We show how to tune both the heap size and the garbage collection of the Java Virtual Machine in order to achieve a good performance while executing a simple locomotion benchmark based on crowd simulations. Furthermore, the paper includes an evaluation of Jason's performance over multi-core processors. The main conclusion derived from this work is that, by means of Java tuning, it is possible to run interactive MAS programmed using Jason.

1 Introduction and Related Work

MAS platforms capable of handling a large amount of complex autonomous agents at interactive response times are required by interactive multiagent applications such as crowd simulations and massive online games. Usually, these kinds of simulations involve a high number of agents (e.g. pedestrians) interacting in a shared environment. Interactivity, in turn, requires the use of parallel techniques that allow to validate and to execute the actions requested within a limited period of time (commonly, 250 ms [6]).

Java-based simulation environments are currently being used by many MAS, since they mainly provide portability as well as an interesting reduction of the development cost. However, this kind of MAS are rarely considered when developing interactive applications with time response constraints, because of Java being normally less efficient than other languages such as C or C++. This situation requests performing a specific Java tuning to be able to tackle this type of

* This work has been jointly supported by the Spanish MEC and the European Commission FEDER funds, under grants Consolider-Ingenio 2010 CSD2006-00046 and TIN2009-14475-C04-04.

J. Li (Ed.): AI 2010, LNAI 6464, pp. 354–363, 2010.

applications. In this paper, we show the Java tuning carried out for the purpose of evaluating the performace of Jason [1], a well-known Java-based MAS platform. The aim of this tuning is to adjust both the heap size and the garbage collection of the Java Virtual Machine in order to satisfy the temporal requirements of interactive multiagent simulations. Therefore, the results presented in this paper will also be of great value to those researches considering Java-based simulation environments suitable for developing interactive multiagent applications.

When developing this kind of interactive MAS three layers are normally considered: the computer architecture, the MAS platform and the graphical engine (if any). At the low level, different distributed computer architectures have been applied in order to allow massive interactive simulations to scale up with the number of agents by simply adding new hardware (e.g. networked-server, P2P, etc.). For instance, a new approach has been presented for PLAYSTATION3 which supports simulation of simple crowds of up to 15000 individuals at 60 frames per second [11]. Parallel simulation, based on classical Reynolds's boids [12], has been also integrated in a PC-Cluster with MPI communication [16] to finally produce small simulations (512 boids). At the top level, the graphical engine of the application must render the visualization at interactive frame rates. The computer graphics community generally represents the MAS as a particle system with local interactions [3,15], though, few works include socially complex and autonomous behaviors [10]. However, they are not normally based on standard agent architectures.

In the middle level, the MAS platform is in charge of providing the required data flow to the graphical engine while efficiently using the computational resources. Thus, it constitutes a key middleware that highly influences the global performance and the scalability of the system. It mainly addresses two important issues: modeling the behavior of the agents as well as their parallel lifecycle execution. Java is a popular language providing built-in support for concurrency that is commonly used by MAS platforms. Although Java performance has been studied from different perspectives, probably the most usual is to tune server applications running on large multi-processor servers [13]. There are more specific works focused on the evaluation of Java-based multiagent platforms [2,14,8]. However, none of them deals with providing interactivity to the corresponding MAS. Some researchers have been also testing the performance and scalability of a few existing MAS platforms [7], showing a lack of both important issues in many of them. In a previous work [5], the authors analysed Jason's architecture and evaluated its performance under both centralised and distributed infrastructures. Regardless the infrastructure, the results showed that the execution options had to be reviewed in order to achieve a more equilibrated response time distribution, an aspect that we have covered in this work.

The rest of the paper is organized as follows. Section 2 briefly reviews Jason's centralised infrastructure and describes the locomotion benchmark used for the evaluation. Section 3 demonstrates how to tune Java in order to run interactive multiagent simulations over Jason. Finally, section 4 shows the performance obtained with different multi-core processors.

2 Test Description

The goal of this work is to evaluate Jason as a suitable framework for running interactive multiagent simulations. Jason is a Java-based interpreter for an extended version of AgentSpeak, a BDI agent-oriented logic programming language [1]. Jason provides three infrastructures to execute a MAS: *Centralised*, SACI and JADE. Whereas the *Centralised* infrastructure places all the components of the MAS in the same host, it is also possible to distribute these components in several hosts using either SACI or JADE technologies. For the sake of simplicity, this paper focuses on the *Centralised* infrastructure but the results obtained are fully applicable for both distributed infrastructures.

In the Jason's *Centralised* infrastructure, the environment has its own execution thread and it is provided with a configurable pool of threads (PThE) devoted to executing the actions requested by the agents. In this way, the enviroment is able to deal with several agent requests concurrently. In turn, each agent owns by default a thread in charge of executing the agent reasoning cycle. In this manner, all the agents can run concurrently within the MAS. As such, this approach could limit the number of agents that can be executed, since the total number of threads would be limited by the Java Virtual Machine (JVM) heap size. However, Jason offers the possibility to optionally add another configurable pool of threads (PThA), so that the set of agents can share a smaller number of execution threads but reducing the level of concurrency. The number of threads in both PThE and PThA is initialised during the start-up of the MAS and it is not changed along its execution. By default, the PThE holds 4 threads whereas the PThA is disabled, so that each agent will have its own execution thread. In a previous work, we tuned both the PThE and the PThA in order to obtain the best performance [5].

The main issue to be tackled when running interactive multiagent simulations is that of being able of efficiently handling a massive and concurrent action processing. In this paper, we have used a locomotion testbed. Here, a set of wanderer agents request movement actions to a grid-like environment, which replies with the result of the execution. Wanderer agents are written in AgenSpeak and they cyclically execute the following steps: (i) take start time, (ii) request a random movement to the enviroment, and (iii) take finish time. On the other hand, the environment executes each movement action in a synchronized manner to ensure the world consistency. That is, the environment performs a simple collision test and informs whether the action can be carried out (i.e. Ok) or it cannot (i.e. Failure), when it would lead to a collision situation.

The performance evaluation carried out along the paper measures the environment response time and the percentage of CPU utilization consumed while running the locomotion benchmark. These measurements represent respectively latency and throughput, the two performance parameters commonly considered when evaluating networked-based distributed computer platforms [4]. We define the Response Time (RT) as the time elapsed between an agent asking for an action and receiving the reply from the environment. Our simulations stop when all the agents have performed 500 movements or cycles, but we discard the first

200 cicles when computing the average response time ($\overline{RT}$). Thus, we measure the system behavior at full load, since the first measurements are distorted due to the agent creation phase.

As stated above, we are interested in exploring the performace of Jason's *Centralised* infrastructure in depth. Thus, both the environment and the agents are run on the same host. The results for the *Centralised* infrastructure shown in [5] indicated that, when simulating 1000 wanderer agents, the 70% of the agents were able to act within 85 ± 264 ms. That is, even though the low value of $\overline{RT}$ (85 ms) indicated that many actions were processed very fast, there were a few agents that must wait more than 250 ms for their actions to be executed. This problem with the high standard deviation of the response time (σ_{RT}), found all over the measures in [5], is addressed in the following section.

3 Java Tuning

The source of the high standard deviation of the response time of Jason-based MAS can be envisoned in figure 1. The figure shows that the average response time per agent cicle ($\overline{RT_c}$) peaks periodically. This points to a process that stops the system whenever it is executed: the Java Garbage Collection. Thus, we have carried out Java Performance Tuning in order to provide some general recommendations for running interactive multiagent simulations over Jason. It should be noticed, though, that the optimal tuning parameters will finally depend on the application and on the hardware underneath.

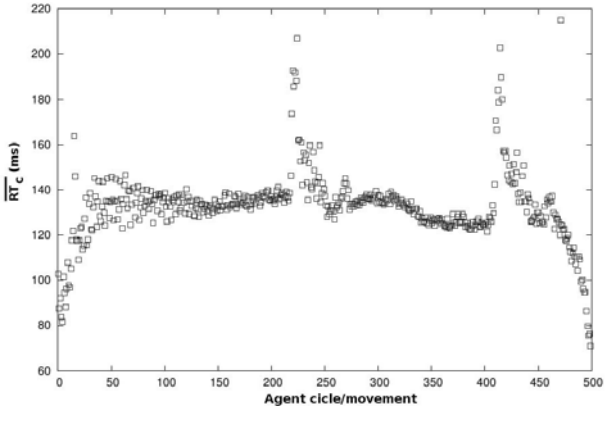


Fig. 1. Influence of the Java Garbage Collection on the response time

In this section, we show the results obtained when executing the testbed defined in section 2 over an AMD Dual-Core Opteron processor with 4 Gb of RAM, running a 64-bit version of Linux and the Sun's HotSpot™ Java Virtual Machine (JVM) release 1.6.0_07. From version 1.5, this JVM has incorporated

a technology to begin to tune itself, referred to as Ergonomics. Even though Ergonomics significantly improves the performance of many applications, optimal results often require manual tuning.

There are two main aspects that have to be tuned in order to enhance Java performance: the heap size and the garbage collector (GC) [9]. Regarding the former, by default, the initial heap size is 1/64th of the machine's physical memory and the maximum heap size is 1/4th of the machine's physical memory. In our case, this would mean using 64 Mb and 1 Gb respectively. However, Java performance can be enhaced by increasing the maximum heap size, as shown in figure 2. This figure shows the total amount of time consumed by the garbage collection when we use diferent GCs and increase the heap size while simulating 2500 agents. This time is computed by adding the times needed to complete every invocation to the GC. Besides, we have set minimum and maximum heap sizes equal for a faster startup. Note how, regardless of the GC being used, the total GC time strongly decreases when increasing the heap size up to 2 Gb. Further on, the gain is very low compared to the fact of being using almost the whole physical memory.

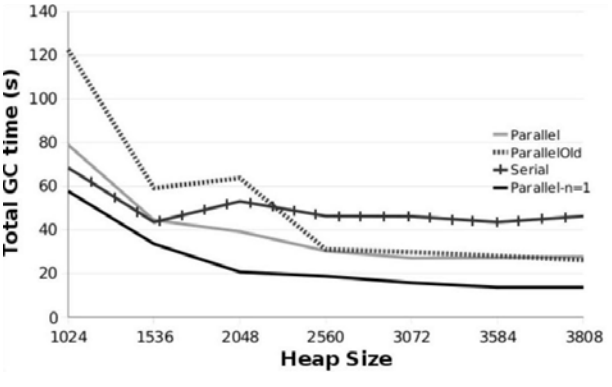


Fig. 2. Garbage colletion time needed for different heap sizes and GCs

With respect to the garbage collectors, Sun's HotSpot^{TM}JVM allows the programmer to choose among three of them: *serial*, *throughput* and *concurrent* low pause collector. Whereas the *serial* GC is a sequential collector, the *throughput* GC uses multiple threads to collect garbage in parallel and it is suitable for applications with a large number of threads allocating objects, such as the one being tested in this paper. On the other hand, the *concurrent* GC does most of the collection concurrently with the execution of the application and it is appropriate for applications that benefit from shorter GC pauses. Additionally, Java GCs organize the object memory into two generations: *young* (recently created objtects) and *tenured* (older objects). Java allows to set the ratio between the *young* and *tenured* generation by means of the JVM command-line option *NewRatio*. For more details on Java garbaje collection, see [9].

Bearing all this informacion in mind, we have executed our benchmark using every GC available. Figure 2 shows the most relevant results that we have obtained. The line named *Serial* corresponds to the total amount of time consumed by the garbage collection when simulating 2500 agents using the *serial* GC. The *Parallel* line relates to the use of the *throughput* GC only for the collection of the *young* generation. In turn, the *ParallelOld* line refers to the use of the *throughput* GC for the collection of both the *young* and the *tenured* generation. For space reasons, we skip the results obtained with the *concurrent* GC since they are up to ten times higher than those obtained with the rest of the GCs, both for the total GC time and for the average response time. As we can observe, the *serial* GC behaves worse than any configuration of the *throughput* GC. Moreover, parallelizing the collection of the *tenured* generation does not fasten but actually slows garbage collection when the heap size is less than 2.5 Gb. This means that there is not a problem with the collection of old objects but with the young ones. The reason behind this fact relies on how Jason represents agent's beliefs and actions. Both are implemented as objects that are discarded and created again whenever there is a change in a belief or a new action is requested to the environment. As each wanderer agent continuously asks the environment for movement actions and changes its position, we can imagine the huge amount of objects that "die young". Thus, enlarging the *young* generation will benefit garbage collection.

The default *NewRatio* for the Server JVM is 2. That is, the *tenured* generation occupies 2/3 of the heap while the *young* generation occupies 1/3. A larger *young* generation could accommodate many more short-lived objects, decreasing the need for slow major collections. Meanwhile, the *tenured* generation would still be large enough to hold many long-lived objects. According to this, the line labeled as *Parallel-n=1* in figure 2 shows that we can obtain the lowest garbage collection times by using the *throughput* GC for the collection of the *young* generation along with the minimum ratio possible between the generations (i.e. $NewRatio = 1$). Hence, half of the heap for the *young* generation and the other half for the *tenured* generation.

Finally, we have evaluated the effect of the number of threads devoted to collect garbage when using the parallel *throughput* GC. By default, this GC uses as many garbage collector threads as the number of processors available. Though, the number of threads can be tuned manually through the *ParallelGCThreads* command-line option. For this test, we have used a 16-core computer and we have varied the number of collector threads from 2 up to 16. Besides, we have tuned Java so it runs efficiently with 2 Gb of heap size and the *NewRatio* set to 1. Figure 3 shows the values obtained for the average response time ($\overline{RT}$) plus its standard deviation (σ_{RT}) when increasing the number of agents simulated. Evidently, the worst values are obtained when only 2 threads are used for garbage collection. However, in our test it is not necessary to use as many threads as the number of cores, since we get the same results for 8 and 16 GC threads.

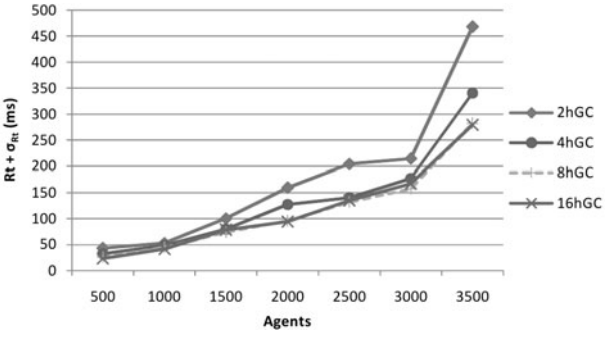


Fig. 3. Performance when varying the number of threads used by the *throughput* GC

Summing up, we can state the following general recommendations for running interactive multiagent simulations over Jason:

- Enlarge the heap size as much as possible without achieving the amount of physical memory available. In addition, set minimum and maximum heap sizes equal for a faster startup.
- Parallelize garbage collection by using the *throughput* GC whenever your hardware has at least 2 CPUs in order to reduce GC pause times. Besides, check whether you need the default number of collector threads (equal to the number of processors) or you can save any, thus reducing the workload of the whole machine.
- Increase the size of the *young* generation up to the size of the *tenured* generation (*NewRatio*=1) to decrease the need for slow major collections.

4 Performance Evaluation

In this section we analyse the results obtained when running the benchmark described in section 2 on the following distributed shared memory (DSM) multicore computers: 2-Core (AMD Dual-Core Opteron, 1.6 GHz, 4 GB RAM), 4-Core (AMD Quad-Core Opteron, 1.0 GHz, 8 GB RAM), 8-Core (Intel 8-Core Xeon, 2.6 GHz, 16 GB RAM) and 16-Core (AMD Dual-Core 8218, 1.0 GHz, 32 GB RAM). All of them run the same 64-bit version of Linux and the Sun's HotSpot^{TM}JVM release 1.6.0_07.

Table 1 shows the performance obtained when simulating from 1500 to 9500 wanderer agents on the computers described above. The results for 1-core were obtained through the *taskset* Linux command. When running the benchmark, we have followed the Java tuning recommendations stated in section 3. Therefore, we have used the *throughput* GC for the collection of the *young* generation with a number of collector threads equal to the number of cores. Besides, we have tuned Java so it runs with 4 Gb of heap size and we have set *NewRatio* to 1. The left column in Table 1 shows the percentage of CPU utilization measured during the execution of the simulation. The central column ($\overline{RT}$) shows the average

Table 1. Performance obtained for Jason framework over different computers

Cores-Agents	CPU(%)	$\overline{RT}$(ms)	σ_{RT}(ms)	Cores-Agents	CPU(%)	$\overline{RT}$(ms)	σ_{RT}(ms)
1-1500	89,53	44,59	101,64	8-1500	59,88	31,01	8,58
1-3500	90,01	40,39	189,57	8-3500	67,75	73,82	22,65
1-5500	89,98	71,97	178,42	8-5500	72,09	114,10	40,62
1-7500	87,87	85,93	193,03	8-7500	74,56	146,27	58,26
1-9500	65,97	98,33	2196,68	8-9500	74,92	185,81	278,00
2-1500	89,17	3,92	28,84	16-1500	39,77	57,38	9,60
2-3500	91,13	5,55	27,59	16-3500	46,45	145,86	38,10
2-5500	92,00	9,01	35,38	16-5500	48,27	242,87	62,23
2-7500	91,10	10,39	79,09	16-7500	57,58	282,57	85,73
2-9500	59,72	47,79	1152,10	16-9500	57,51	253,53	534,66
4-1500	76,25	51,97	20,81				
4-3500	81,11	132,88	50,71				
4-5500	81,48	201,90	76,89				
4-7500	83,35	290,71	118,30				
4-9500	84,24	386,35	488,37				

Response Time in milliseconds for the actions requested by the agents when the system is at full load, as explained in section 2. Finally, the right column shows the standard deviation of this Response Time (σ_{RT}).

The results shown in Table 1 demonstrate that we can run interactive multiagent simulations over Jason, since the values of the $\overline{RT}$ plus the σ_{RT} are generally under the reference value of 250 ms. As it was also expected, the CPU utilization decreases as the number of cores increases. For instance, if we compare the results obtained for 3500 agents on each computer, it can be seen that the more cores in the computer, the lower the percentage of CPU utilization (the single CPU is shown only as a reference). However, the response time does not behave the same way. Instead, whereas the $\overline{RT}$ values for the 2-Core computers are around a few milliseconds, the $\overline{RT}$ for the computers with 4 up to 16 cores reaches tens of milliseconds. The worsening of the response time occurs in all the computer being tested, although it has a minor impact in the 8-Core computer because it has the highest processor speed. This fact indicates that, beyond two cores, the default configuration used by Jason does not properly scale up with the number of processor cores. Thus, a deeper study must be carried out in order to allow it to take advantage of the multi-core processors.

Although a fine tuning of the Jason framework for multi-core processors is beyond the scope of this paper, we have analysed the issue shown in Table 1 in order to clarify the path for future work. We think that the reason behind this problem is thread context switching. Even though the Java Virtual Machine schedules its threads to run them as fast as possible, there is no guarantee of which core a given thread will be assigned to for execution. The operating system kernel can assing one single thread to different cores during its execution time, thus provoking thread migrations. The probability of migration increases with the number of cores in the processor, in such a way that the overhead due to

thread migrations could exceed the benefits of having more cores for executing the threads in parallel. To verify this hypothesis, we have measured the number of migrations (i.e. changes in the core assigned for execution) suffered by the threads along the simulation. To detect migrations, we have used a system call retrieving the state of the Java threads periodically and we have analysed the core where they were located.

Figure 4 shows the total number of thread migrations counted while executing the same simulations that produced the results of Table 1. We can observe how the number of migrations is proportional to the number of cores in the computer. Since a thread migration is a time consuming task, the high number of migrations produced by computers with more than 2 cores can explain the behavior shown in Table 1. Nevertheless, it should be noticed that these results do not guarantee the absence of other still hidden aspects that could prevent the system from properly scaling with the number of processor cores. In order to fully exploit the degree of parallelism offered by multi-core processors, tuning the processor affinity of Jason must be done.

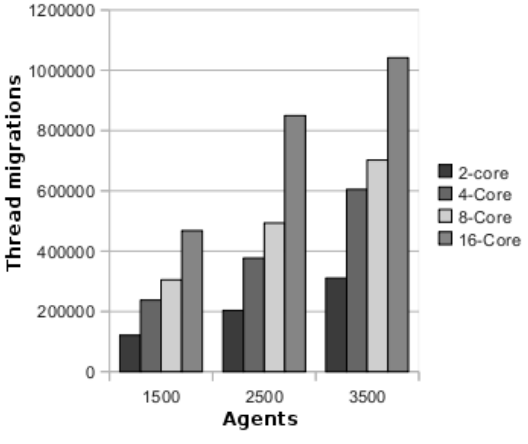


Fig. 4. Number of thread migrations

5 Conclusions and Future Work

In this paper, we have evaluated Jason as a suitable Java-based MAS platform for developing interactive multiagent simulations. We have shown how to tune the Java heap size as well as the gargabe collector in order to enhance the performance of the simulations. Even though the optimal tuning parameters will finally depend on the application and on the hardware underneath, we have state some general recommendations for minimizing the impact of garbage collection. Therefore, the results presented in this paper will also be of great value to those researches considering other Java-based simulation environments for developing interactive multiagent applications. The paper also includes a first evaluation

of Jason's performance over multi-core processors. As future work, we plan to carry out a deep study of the Jason framework in order to properly scale it up with the number of processor cores. Then, tuning the Java processor affinity will be required to exploit the degree of parallelism offered by multi-core processors.

References

1. Bordini, R.H., Hübner, J.F., Wooldrige, M.: Programming Multi-Agent Systems in AgentSpeak using Jason. Wiley, Chichester (2007)
2. Cortese, E., Quarta, F., Vitaglione, G.: Scalability and performance of JADE message transport system. In: AAMAS Workshop on AgentCities (2002)
3. Dobbyn, S., Hamill, J., O'Conor, K., O'Sullivan, C.: Geopostors: a real-time geometry/impostor crowd rendering system. ACM Trans. Graph. 24(3), 933–933 (2005)
4. Duato, J., Yalamanchili, S., Ni, L.: Interconnection Networks: An Engineering Approach. Morgan Kaufmann, San Francisco (2002)
5. Fernández, V., Grimaldo, F., Lozano, M., Orduña, J.M.: Evaluating Jason for distributed crowd simulations. In: Proc. of the 2nd International Conference on Agents and Artificial Intelligence, vol. 2, pp. 206–211. INSTICC Press (2010)
6. Lozano, M., Morillo, P., Ordua, J.M., Cavero, V.: On the design of an efficient architercture for supporting large crowds of autonomous agents. In: Proceedings of IEEE 21th International Conference on Advanced Information Networking and Applications (AINA 2007), pp. 716–723 (2007)
7. Mulet, L., Such, J.M., Alberola, J.M.: Performance evaluation of open-source multiagent platforms. In: Proc. of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1107–1109. ACM, New York (2006)
8. Nikolai, C., Madey, G.: Tools of the trade: A survey of various agent based modeling platforms. Journal of Artificial Societies and Social Simulation 12(2), 2 (2009)
9. Oracle Sun Depeloper Network. Java Tuning White Paper, 2010, `http://java.sun.com/performance/reference/whitepapers/tuning.html`
10. Pelechano, N., Allbeck, J.M., Badler, N.I.: Virtual crowds: Methods, simulation, and control. Synthesis Lectures on Computer Graphics and Animation 3(1), 1–176 (2008)
11. Reynolds, C.: Big fast crowds on ps3. In: Proc. of the 2006 ACM SIGGRAPH Symposium on Videogames, pp. 113–121. ACM, New York (2006)
12. Reynolds, C.W.: Flocks, herds and schools: A distributed behavioral model. In: SIGGRAPH 1987: Proc. of the 14th Annual Conference on Computer Graphics and Interactive Techniques, pp. 25–34. ACM, New York (1987)
13. Shirazi, J.: Java Performance Tuning. O'Reilly, Sebastopol (2003)
14. Tobias, R., Hoffman, C.: Evaluation of free java-libraries for social-scientific agent based simulation. Journal of Artificial Societies and Social Simulation 7(1) (2004)
15. Treuille, A., Cooper, S., Popovic, Z.: Continuum crowds. In: SIGGRAPH 2006: ACM SIGGRAPH 2006 Papers, pp. 1160–1168. ACM, New York (2006)
16. Zhou, B., Zhou, S.: Parallel simulation of group behaviors. In: WSC 2004: Proceedings of the 36th Conference on Winter Simulation. Winter Simulation Conference, pp. 364–370 (2004)

Emergence of Cooperation in a Bio-inspired Multi-agent System

Hiroyuki Ishiwata, Nasimul Noman, and Hitoshi Iba

Department of Electrical Engineering and Information Systems,
Graduate School of Engineering, University of Tokyo, Japan
{ishiwata,noman,iba}@iba.t.u-tokyo.ac.jp

Abstract. Cooperative behavior of social insects is widely studied and mimicked in Artificial Intelligence communities. One such interesting cooperation is observed in the form of philanthropic activity e.g. army ants build bridges using their own bodies along the route from a food source to the nest. Such altruistic behavior helps to optimize the food gathering performance of the ant colony. This paper presents a multi-agent simulation inspired by army ant behavior. Such cooperation in a multi agent system can be very valuable for engineering applications. The purpose of this study is to model and comprehend this biological behavior by computer simulation.

1 Background

The morphology and behavior of organisms in nature have evolved over a very long period of time. The organisms that have skillfully adapted to the environment have survived until the present. Therefore, the forms and behavior of these organisms have been optimized over the centuries and many of these adaptations are applicable in engineering [14,5,12,10,3]. In particular, social insects such as ants and bees form miniature societies within their nests and exhibit very effective cooperation [15,6]. Ant colony optimization [7] is one of the most famous applications of social insect to engineering.

Recently, several studies have been made on Constructive Approach [8,11,16,9]. Constructive Approach is a kind of reverse engineering. This approach imitates a model to understand the object such as an actual living thing or a nature system. In this study, we created a computer stimulation to model and understand the altruistic behavior observed in army ants during foraging. Successful biomimicry of such behavior of ants can find valuable engineering applications.

2 Altruism of Army Ants

Altruism refers to behavior that prioritizes benefits to others rather than self and sometimes involves acts of self-sacrifice in order to aid others. Some army ants construct living bridges with their own bodies when they find holes or gullies as obstacles to their marching routes as shown in Fig. 1 [4]. Such philanthropic acts

J. Li (Ed.): AI 2010, LNAI 6464, pp. 364–374, 2010.

Fig. 1. Scene of building living bridge by army ant

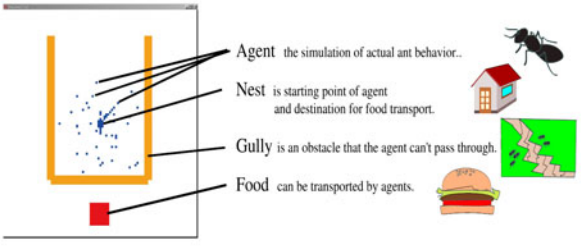


Fig. 2. Simulation environment

are different from the regular behavior of the ants e.g. foraging for and transport of food. However, if more ants participate in bridge construction than that is required or if they construct bridges at sites where those are unnecessary, they may actually hamper the food gathering performance of the whole colony. But, in nature, the ants are very keen to balance these actions as per requirement and it has been confirmed that because of such altruistic activity the performance is improved for the group as a whole. In an experiment by Powell *et al.*, it was found that the foraging capacity of the army-ant colony increased by up to 26% due to this altruistic behavior [13]. In this study, this altruism of ants is modeled and examined in a multi-agent simulation environment.

3 Defining the Problem

This section explains the problems handled in the multi-agent simulation. The present simulation serves a model for the foraging behavior and the previously described altruism of ants. The simulation was performed using Swarm library. Fig. 2 shows a screenshot of the simulation screen where an agent represents an ant movement.

The actions include foraging for and transport of food and communications with neighboring ants using pheromone. The nest is the starting point of the agents and also the point to which the agents return with food. The pheromone is released by an agent when it finds food. Just as in nature, once secreted

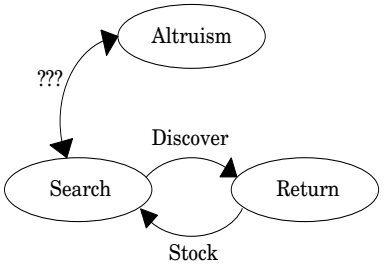


Fig. 3. State transition of agents

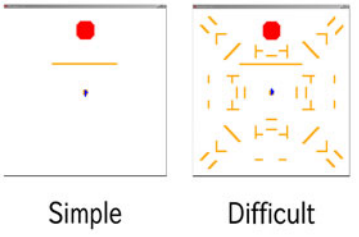


Fig. 4. Maps for experiment

Table 1. States and behavior of agents

State	Behavior
Search	This is the initial condition of the agent and it continues random work until food is found. When food is found, there is a transition to the Return state. Transition to Altruism state is also possible under "certain" conditions. When pheromone is sensed, the ants are drawn to the higher concentrations.
Return	The food is returned to the nest. In this state the agent moves toward nest secreting pheromone. After reaching the nest, the agent transits to the Search state. An agent in Return state knows the position of the nest.
Altruism	A bridge is constructed across the gully. While in this state, movement is impossible for an agent. When certain conditions are met, the bridge is abandoned and the agents transit to the Search state.

the pheromone attenuate and disperse, thus disseminating information among the ants about the locations of food. A gully hinders movement of agents and fundamentally prevents the agents from passing over it. However, if an agent shows altruism and forms a living bridge over the gully, other agents can pass over the gully. The agents move in accordance with the state transition diagram shown in Fig. 3. The behavior of agents in different states are shown in Table 1.

The problem is to determine the conditions that induce the transition to the Altruism state. But it is not concretely known how ants decide the site and timing of living-bridge construction and when they cease the bridge formation. Therefore, in this study, several hypotheses are proposed as the altruism initiation conditions and experiments were performed for verification.

4 Judgment Criteria for Entering Altruism State

4.1 Hypotheses

Here, two hypotheses were proposed as the judgement criteria for altruistic activity by army ants.

Table 2. Properties used in Model 1 & 2

	Model 1	Model 2
Number of Step	700	700
Time	10	150
Radius	2	-
Pheromone Threshhold	-	30

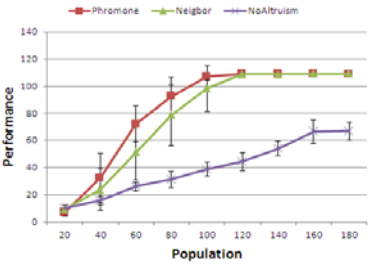


Fig. 5. Simple map - Experimental results

Model 1: Based on the Presence of Neighboring Ants

An ant will start formation of living bridge over a gully only when neighboring ants are present. Hypothetically, this approach will be more efficient compared to forming a bridge blindly because when there are neighboring ants the probability is high that they will utilize the shortcut.

Model 2: Based on the Presence of Pheromone

As described earlier, agents secrete pheromone when they find food, and this pheromone is used to disseminate information among the ants about the location of the food. Therefore, the places where pheromone concentrations are higher than a fixed level are the locations that many ants have passed and/or will pass through in the future. Hence, a living bridge can be formed judging the pheromone concentration.

In both models, agents leave the bridge after a fixed amount of time passes. And we used fixed properties optimized by genetic algorithms (Shown in Table 2). In order to judge their validity, these hypotheses were fed into the simulation and their usefulness was verified empirically.

4.2 Experiment to Verify the Hypotheses

The two scenarios shown in Fig. 4 were used in the experiment. In these experiments, performance was measured using the number of food items collected within a fixed period of time. Each experiment was repeated 10 times with 20 to 180 agents, increased by 20 at a time, and the mean values were compared.

The experimental results from the simple map are shown in Fig. 5. The numbers of agents is shown along the horizontal axis and the number of food items collected within a fixed amount of time is shown along the vertical axis. In simple map, the Model 1 showed slightly higher performance, but the differences were small and almost no difference in overall efficiency was observed.

Experimental results using the difficult map are shown in Fig. 6 and Fig. 7. On the whole, Model 2 performed better in the difficult map. Fig. 7 shows experimental observations for the difficult map on a different scale. Just like before, the horizontal axis represents the number of agents, however, the vertical

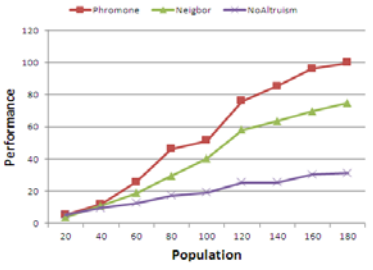


Fig. 6. Difficult map - Experimental results 1

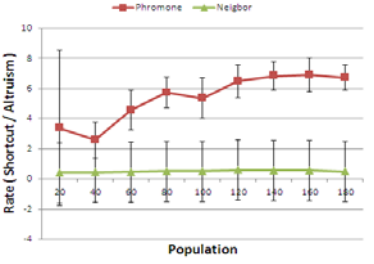


Fig. 7. Difficult map - Experimental results 2

axis represents the ratio of the total number of times agents crossed bridges to the total number of times agents helped to form bridges. This ratio indicates how useful the formed bridges were. From the data, it was found that Model 2 yielded higher values than Model 1. For Model 1, the ratio was usually about one. This means that even though a bridge was formed, neighboring agents would have not used it efficiently. This was because in the difficult map, unlike the simple map, gullies were present at various locations causing bridges to be formed at unnecessary sites with Model 1. With Model 2 higher ratios were found compared to that found with Model 1. Although it is not evident from the graph, in Model 2 the bridges were formed only at those sites that was necessary for bringing food to the nest. This was because the pheromone was secreted along the way from the food-source to the nest. And the concentration of pheromone indicated the the optimal sites for bridge construction. Hence, both the timing and sites of bridge construction were superior in Model 2. However, Model 2 suffers from the drawback that bridges cannot be formed until the foraging sites have been found. In nature, cases are also observed where bridges are formed at necessary sites before foraging sites are found. So we hypothesize that, for altruistic activity like bridge formation ants use the pheromone method along with some other judgement criteria such as the one stated in Model 1.

5 Judgment Criteria with Reference to Chain Formation

5.1 What Is Chain Formation?

Chain formation is another philanthropic cooperative behavior similar to bridge formation. Chains in this case refer to structures formed by the bodies of the ants when the ants encounter extreme differences in heights during their marches. In this way, it is possible for other ants to move safely from one height to another. In their research, Lioni *et al.* [1] observed the chain formation behavior of ants in nests installed in the laboratory. The results showed that the probability of participation in chain formation P_e and probability of abandoning chain formation P_s can be approximated by the following equations:

$$P_e = C_{e0} + \frac{C_{e1}X}{C_{e2} + X} \qquad (1) \qquad\qquad P_s = C_{s0} + \frac{C_{s1}X}{C_{s2} + X^{\nu}}, \qquad (2)$$

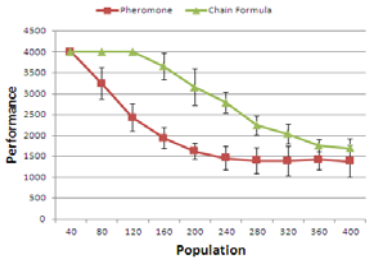


Fig. 8. Performance comparison in terms foraging time

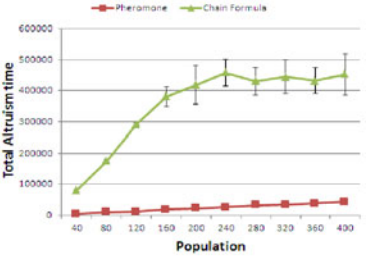


Fig. 9. Performance comparison in terms of altruistic activity

where X is the number of ants participating in chain formation and the other numbers are constants. According to these equations, if many ants are contained in the formed chain then it is easier for them to participate in chain formation but more difficult for them to stop. Using these formulas as judgment criteria for chain formation an experiment was done.

5.2 Experiment to Verify the Chain Formation System

To justify the proposed model of pheromone concentration as the criteria for transition to altruism state, a comparative study was performed with the Lioni's model of chain formation.

In Fig. 8, the number of agents is shown on the horizontal axis and the time until completion of foraging on the vertical axis. It was found that for some population sizes when pheromone concentration is used as judgment criteria, foraging takes shorter time than that is required for chain formula.

In Fig. 9, the number of agents is shown on the horizontal axis and the cumulative time during which the agents are engaged in altruistic behavior on the vertical axis. It was also observed that when pheromone concentration is used as judgment criteria, the total time during which the agent are engaged in altruistic activity is shorter and less affected by the population size. On the other hand when the formulas of Lioni *et al* are applied, the time engaged in altruistic behavior increases with the number of agents. Fig. 10 compares another aspect of the models. When pheromone concentration was used as judgment criteria, bridges were constructed at the required sites, but when the formulas of Lioni *et al.* were applied, bridges were constructed at many sites other than the required sites. It is also clear from Fig. 10 that with the Lioni *et al.* model, fewer numbers of agents are in Search state as many of them are in Altruism state.

Procedures using formula (1) of Lioni *et al.* featured a higher probability of altruistic behavior at sites where agents are apt to congregate. Therefore, more altruistic behavior is expected to occur close to the foraging site and the nest or in between these sites. In Lioni's model, the altruistic behavior is possible without finding foraging sites and this is an advantage over the proposed model based on pheromone concentrations. Nevertheless, the simulation results showed that in terms of performance, measured as foraging speed, the proposed model

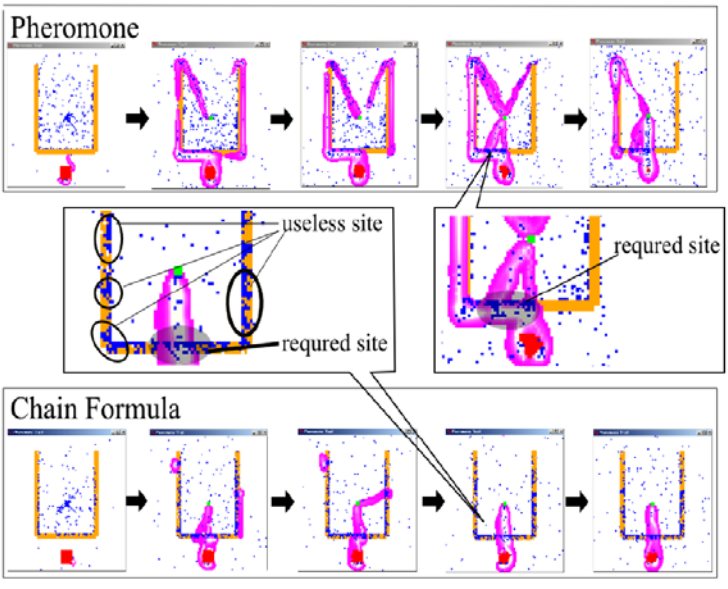


Fig. 10. Comparison of bridge construction sites

was superior to Lioni's model. The possible reason behind this could be that in Lioni *et al.* experiment calculations were performed by limiting the chain formation sites to one, hence, their model could not be directly applied to an environment with a series of bridge formation sites as used here. Therefore, in consideration of biology, etc. of army ants, we need to combine the pheromone concentration based model with other judgment criteria.

6 Changes in Strategy Based on Numbers of Agents

6.1 Deciding Group Behavior of Army Ants

It has been confirmed that the group behavior of army ants is seriously affected by the number of ants that are active [1,2]. For example, when few ants are available for chain formation, chains are not formed but when large numbers of ants are available, chains are formed at several sites. However, when the number of active ants is moderate, initially several chains are formed. But after a certain time, extension of most of the chains stops and the chains gradually decrease in size and eventually the extension of only one chain continues. However, it is still not clear how the ants count the number of neighboring ants and how this number affects their behavior.

6.2 Comparative Experiment

In order to monitor the effect of group size on the activity of agents, we performed experiments using Lioni *et al.* formulas extended with a minimum limit on group

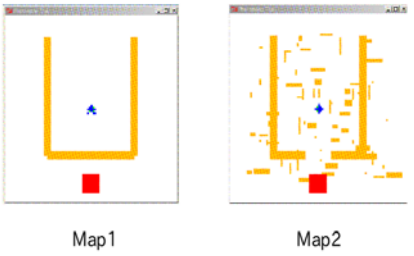


Fig. 11. Maps used to study the effect of number of agents

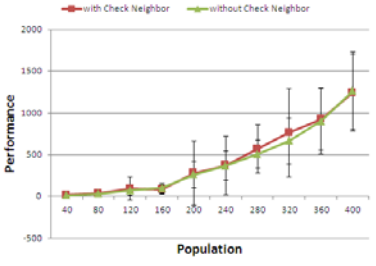


Fig. 12. Effect of neighborhood knowledge (Map 1)

Fig. 13. Effect of neighborhood knowledge (Map 2)

size as an additional condition of chain formation. We compared this scheme with the one that does not take into account the group size. The experiment was performed using two maps shown in Fig. 11. The results of the experiments are shown in Fig. 12 and Fig. 13. The horizontal axis shows the number of agents, and the vertical axis shows the performance in terms of the number of food items collected within a fixed time. In these figures, "with Check Neighbor" represents the procedure taking the number of neighboring ants into consideration and "without Check Neighbor" indicates the procedure not taking the number of neighboring ants into consideration

In Map 1, the method that did not take into account information about neighboring ants showed high performance. This was because the conditions for bridge formation were relaxed and hence bridges could be formed at an early stage and food can be found easily.

Map 2 was used to investigate whether intelligent behavior can be achieved by avoiding unnecessary bridge formation where a shortcut is not especially necessary for food collection. In this case, better results were obtained with the method that checks the number of neighboring ants.

Fig. 14 shows how the bridges extend in size with time for Map 1. In the figure, 1st refers to the largest bridge at the time and 2nd to the next largest bridge. The horizontal axis shows time and the vertical axis the two largest bridges. It is apparent from the graph that at first, several bridges coexist and extend for about the same length, but finally the differences become greater. Fig. 15 shows

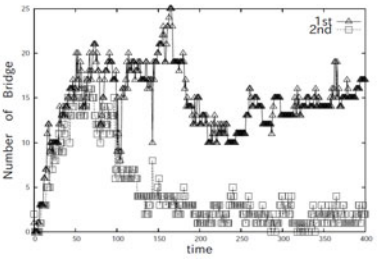


Fig. 14. Changes in size of bridge

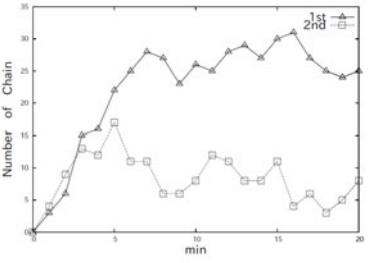


Fig. 15. Changes in size of chain (Data plot of [2])

the data obtained in a biological experiment in the research of Lioni *et al.* [2]. When chains were formed at two sites, records were kept on how each of the chains extended. In the figure 1st and 2nd show the sizes of the chains at each site. Fig. 15 was newly plotted based on data taken from the paper.

7 Simulation with Fixed Role Assigned

From the previous experiments, it seems that our model has many properties similar to actual army ant behavior. To emphasize the similarity between the simulator agents and the actual army ants, it is important to compare experimental data. We can do that by corresponding the agents behavior to the army ants behavior.

As a first step, the experiment was performed using a simulator that has agents with fixed task assigned. Task assignment is the one of the signatures that is observed in army ant. Army ants have tasks that depend on someone's rank. Here we consider two different roles for agents in our simulator

- role A : Search and transport food.
- role B : Build a bridge to support role A.

We performed experiment by assigning agents in these two roles with different ratios. Fig. 16 shows the experimental results where the performance was compared in terms of the number of food items collected within a fixed time. Rate 0.1 means that 10% agents were assigned to role B in the simulation. "Dynamic Assignment" labels the experimental results obtained by the simulator used in Section 6 where the agents have no fixed role.

The results in Fig. 16 indicate that a fixed division of roles may be better than a dynamic one. Especially Rate 0.5 and Rate 0.6 are better than other ratios. Although "Dynamic Assignment" was not the best, it performed competitively on an average. This study points out the possibility of role assignment in our simulator. But, in the real world, it is not possible to know the role assignments of ants to solve this problem. Nevertheless, a broader survey can help us to improve our simulation.

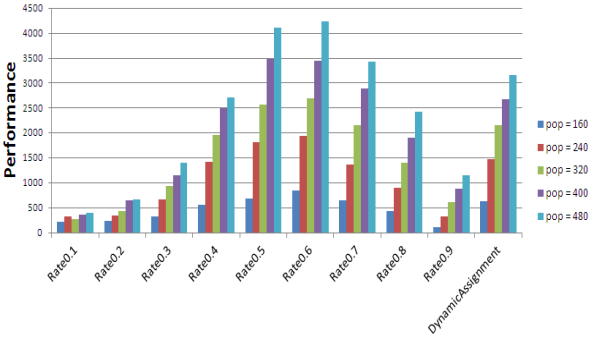


Fig. 16. Experimental results with task assignment

8 Conclusion

In this work, we have studied a couple of models to simulate the altruistic behavior by army-ants for evolving cooperation in a multi-agent environment. We observed that timely and efficient formation of live-bridges is possible using the pheromone concentration as the condition for altruism; but for that a foraging site had to be found beforehand. Since the actual actions of ants are not restricted in this way, the chain formation probability by Lioni *et al.* was used as the judgment criteria for bridge formation. However, differences appeared between the simulated environment and what is observed in nature; and the foraging performance of the ant colony was decreased.

Then we experimented using the chain formation probability, along with the number of neighboring active ants, as the condition for altruistic behavior. In this case, bridge formation at unnecessary sites is decreased and performance improved. When the changes in the size of the bridges are considered, the behavior by actual ants that enables them to make collective decisions has been observed. However, since there are also cases when higher performance was possible without considering the numbers of neighboring ants, there is still room for improvement in this respect.

In order to make the simulation environment more realistic, experiments with fixed role assigned to the agents were performed. Experiments with roles assigned in certain ratios showed high performance. But role assignment mechanism in real world army ants is unknown.

This study shows that mimicking the altruistic behavior in army ants, it is possible to evolve improved and efficient cooperation in a multi-agent environment. Such effective cooperation can be used to solve many difficult real world problems ranging from robotics to computer games. However, there is plenty of scope for improving the proposed models incorporating more concrete knowledge about army ants' behavior in nature.

References

1. Lioni, A., Sauwens, C., Theraulaz, G., Deneubourg, J.-L.: Chain formation in *oecophylla longinoda*. Journal of Insect Behavior 14(5), 679–696 (2001)
2. Lioni, A., Deneubourg, J.-L.: Collective decision through self-assembling. Naturwissenschaften 91(5), 237–241 (2004)
3. Azzag, H., Monmarche, N., Slimane, M., Venturini, G.: Anttree: a new model for clustering with artificial ants. In: IEEE Congress on Evolutionary Computation, vol. 4, pp. 2642–2647 (2003)
4. Anderson, C., Theraulaz, G., Deneubourg, J.L.: Self-assemblages in insect societies. Insectes Sociaux 49(2), 99–110 (2002)
5. Teodorovic, D.: Swarm intelligence systems for transportation engineering:principles and applications. Transportation research Part C: Emerging Technologies 16(6), 651–667 (2008)
6. Deneubourg, J.L., Lioni, A., Detrain, C.: Dynamics of aggregation and emergence of cooperation. The Biological Bulletin 202(3), 262–267 (2002)
7. Dorigo, M., Caro, G.D., Gambardella, L.M.: Ant algorithms for discreate optimization. Artifical Life 5(2), 137–172 (1999)
8. Iwasa, Y., Higashi, M., Yamamura, N.: Prey distribution as a factor determining the choice of optimal foraging strategy. The American Naturalist 117(5), 710–723 (1981)
9. Metivier, M., Lattaud, C., Heudin, J.-C.: A stress-based speciation model in lifedrop. In: Artificial life VIII: Proceedings of the Eighth International Conference on Artificial Life, pp. 121–126 (2003)
10. Payton, D., Estkowski, R., Howard, M.: Compound behaviors in pheromone robotics. Robotics and Autonomous Systems 44, 229–240 (2003)
11. Pollack, J.B., Lipson, H.: The golem project: Evolving hardware bodies and brains. In: The Second NASA/DoD Workshop on Evolvable Hardware, EH 2000 (2000)
12. Purnamadjaja, A.H., Russell, R.A.: Guiding robots' behaviors using pheromone communication. Auton. Robots 23(2), 113–130 (2007)
13. Powell, S., Franks, N.R.: How a few help all: living pothole plugs speed prey delivery in the army ant *Eciton burchellii*. Animal Behaviour 73(6), 1067–1076 (2007)
14. von Mammen, S., Christian, J.: Evolutionary swarm design of architectural idea models. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, pp. 143–150. ACM, New York (2008)
15. Holldobler, B., Wilson, E.O.: The multiple recruitment system of the african weaver ant *oecophylla longinoda*. Behavioral Ecology and Sociobiology 3(1), 19–60 (1978)
16. Yamaguchi, M., Yoshimoto, E., Kondo, S.: Pattern regulation in the stripe of zebrafish suggests an underlying dynamic and autonomous mechanism. Proc. Natl. Acad. Sci. USA 104(12), 4790–4793 (2007)

An Efficient Approach for Ordering Outcomes and Making Social Choices with CP-Nets

Minyi Li, Quoc Bao Vo, and Ryszard Kowalczyk

Faculty of ICT, Swinburne University of Technology
John Street, Hawthorn, VIC 3122, Australia
{myli,Bvo,RKowalczyk}@swin.edu.au

Abstract. In classical decision theory, the agents' preferences are typically modelled with utility functions that form the base for individual and multi-agent decision-making. However, utility-based preference elicitation is often complicated and sometimes not so user-friendly. In this paper, we investigate the theory of CP-nets (conditional preference networks) as a formal model for representing and reasoning with the agents' preferences. The contribution of this paper is two-fold. First, we propose a tool, called RA-Tree (Relational Assignment Tree), to generate the preference order over the outcome space for an individual agent. Moreover, when multiple agents interact, there is a need to make social choices. But given a large number of possible alternatives, it is impractical to search the collective optimal outcomes from the entire outcome space. Thus, in this paper, we provide a novel procedure to generate the optimal outcome set for multiple agents. The proposed procedure reduces the size of the search space and is computationally efficient.

Keywords: Social Choice; CP-nets; Collective decision-making.

1 Introduction

Classical decision theory represents agents as having preferences over the outcome space, and the preferences are usually mathematically represented by utility functions. Unfortunately, in many situations, the utility-based preference elicitation is complicated and typical users may not be able to provide much more than qualitative rankings of outcomes [3]. For instance, in everyday banking situation, it is difficult to require people to define their utility functions over multiple issues (e.g. interest rate, annual fee, minimum deposit, choice of terms, etc.) before they apply for a term deposit service. But it is easy for them to provide some statements like "I prefer the no-fee account to the one with monthly administration fees". Moreover, conditional preferences are easier to represent in a qualitative way rather than by quantitative preference. For instance, "I prefer to choose a longer term deposit (e.g. 24 months) if there is no account fee and the current interest rate is higher than 6%".

In this paper, we investigate the theory of CP-nets [2] as a formal model for representing and reasoning with the agents' preference. The motivation of applying CP-nets is that, CP-nets can be used to specify individual preference relations in a relatively

J. Li (Ed.): AI 2010, LNAI 6464, pp. 375–384, 2010.

compact, intuitive, and structured manner, making it easier to encode human preferences and support decision-making in real world applications. We also propose a new technique, called RA-Tree, to reason with CP-net topologically, and generate the preference order over the outcome space for an individual agent.

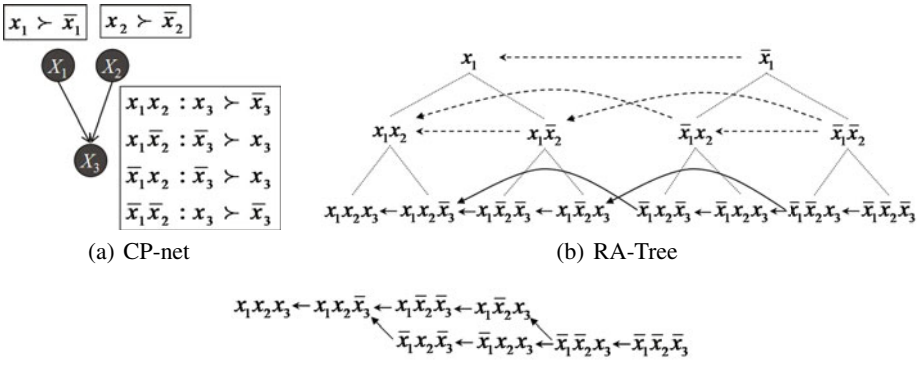
When multiple agents interact, there is a need to make social choices [1]. In this paper, given that the individual preferences have been elicited and represented as CP-nets, the problem of computing the optimal outcome set for multiple agents will be addressed. Recent work on the complexity of computing dominance relations shows that dominance testing[1] for an arbitrary CP-net is PSPACE-complete [4]. However, outcome optimization with multiple agents' CP-nets furthermore requires dominance testing on each pair of alternatives on each individual CP-net. For example, having 10 binary variables, each involved agent would need to compare $\binom{2^{10}}{2} = 523776$ pairs of alternatives. This problem is likely to be even harder than NP or coNP problems. Group decision-making with CP-nets has been studied in the literature, e.g.,[5,9]. However, most existing works either do not consider computational requirements, or depend on a strong assumption that the agents' preferences extend acyclic CP-nets being compatible with a common order on the variables. Other related works are Li *et al.* [6] and Li *et al.* [7]. The former applies majority rule on arbitrary profiles and proposes an algorithm to find the possible majority-optimal alternatives; the latter employs a heuristic to reduce the size of the search space and tries to approximate the fair and optimal outcomes.

To this end, this paper aims at generating the set of all Pareto-optimal outcomes from a collection of CP-nets. The proposed procedure iteratively deletes the dominated outcomes by traversing the agents' RA-Trees. It allows the agents to have different preferential independence structures, guarantees optimality and is computationally efficient. Especially when the agents' CP-net structures are similar in topology, it prunes the dominated assignments efficiently from the high level in the RA-Trees and greatly reduces the search effort. The paper is structured as follows. We first provide background information on CP-net in Section 2. Then we discuss how to reason with CP-net topologically and present the proposed method to generate the preference order over the outcome space for an individual agent in Section 3; based on this, we present the proposed procedure to generate the optimal outcome set for multiple agents in Section 4. Finally we present the concluding remarks and discuss the future work in Section 5.

2 CP-Net Overview

Let $\mathbf{V} = \{X_1, \ldots, X_n\}$ be a set of n variables. For each $X_k \in \mathbf{V}$, $D(X_k)$ is the *value domain* of X_k. If $\mathbf{X} = \{X_{i_1}, \ldots, X_{i_p}\} \subseteq \mathbf{V}$, with $i_1 < \cdots < i_p$ then $D(\mathbf{X})$ denotes $D(X_{i_1}) \times \cdots \times D(X_{i_p})$ and $\mathbf{x}$ denotes an assignment of variable values to $\mathbf{X}$ ($\mathbf{x} \in D(\mathbf{X})$). If $\mathbf{X} = \mathbf{V}$, $\mathbf{x}$ is a *complete assignment*; otherwise $\mathbf{x}$ is called a *partial assignment*. If $\mathbf{x}$ and $\mathbf{y}$ are assignments to disjoint sets $\mathbf{X}$ and $\mathbf{Y}$, respectively ($\mathbf{X} \cap \mathbf{Y} = \emptyset$), we denote the combination of $\mathbf{x}$ and $\mathbf{y}$ by $\mathbf{xy}$. For any assignment $\mathbf{x} \in D(\mathbf{X})$, we denote by $\mathbf{x}[X_k]$ the value $x_k \in D(X_k)$ ($X_k \in \mathbf{X}$) assigned to variable X_k by that

[1] A dominance testing, given an individual CP-net and two alternatives o and o', tests whether o is preferred to o' according to the preferences induced by that CP-net.



(a) CP-net (b) RA-Tree

(c) Preference order over the outcome space

Fig. 1. An example of CP-net, its RA-Tree and preference order over the outcome space

assignment. Let $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ be nonempty sets that partition $\mathbf{V}$ and $\succ$ a preference relation over $D(\mathbf{V})$. $\mathbf{X}$ is *(conditionally) preferentially independent* of $\mathbf{Y}$ given $\mathbf{Z}$ iff for all $\mathbf{x}, \mathbf{x}' \in D(\mathbf{X})$, $\mathbf{y}, \mathbf{y}' \in D(\mathbf{Y})$, $\mathbf{z} \in D(\mathbf{Z})$:

$$\mathbf{xyz} \succ \mathbf{x}'\mathbf{yz} \text{ iff } \mathbf{xy}'\mathbf{z} \succ \mathbf{x}'\mathbf{y}'\mathbf{z}$$

A CP-net $\mathcal{N}$ [2] over $\mathbf{V}$ is an annotated directed graph $\mathcal{G}$ over $X_1, \ldots, X_n$, in which nodes stand for the problem variables. Each node X_k is annotated with a conditional preference table $CPT(X_k)$, which associates a total order $\succ^{X_k|\mathbf{u}}$ with each instantiation $\mathbf{u}$ of X_k's parents $Pa(X_k)$, i.e. $\mathbf{u} \in D(Pa(X_k))$. For instance, let $\mathbf{V} = \{X_1, X_2, X_3\}$, all three being binary, and assume that the preference of a given agent over all possible outcomes can be defined by a CP-net whose structural part is the directed acyclic graph $\mathcal{G} = \{(X_1, X_2), (X_1, X_3), (X_2, X_3)\}$; this means that the agent's preference over the values of X_1 is unconditional, preference over the values of X_2 (resp. X_3) is fully determined given the values of X_1 (resp. the values of X_1 and X_2). The preference statements contained in the conditional preference tables are written with the usual notation, that is, $x_1\bar{x}_2 : \bar{x}_3 \succ x_3$ means that when $X_1 = x_1$ and $X_2 = \bar{x}_2$ then $X_3 = \bar{x}_3$ is preferred to $X_3 = x_3$. Figure 1(a) shows an example of an agent's CP-net.

In this paper, we make the classical assumption that each agent's CP-nets is acyclic and do not model indifference, i.e. the ordering $\succ^{X_k|\mathbf{u}}$, $\mathbf{u} \in D(Pa(X_k))$, expressed in the conditional preference table is total. An individual CP-net $\mathcal{N}$ induces a preference ranking over the outcome space [2]: for any pair of outcomes o_1 and o_2, $\mathcal{N}$ entails $o_1 \succ o_2$ (denoted by $\mathcal{N} \models o_1 \succ o_2$) iff there exists a sequence of *improving flips* from o_2 to o_1 respecting the conditional preference tables of $\mathcal{N}$; otherwise, $\mathcal{N} \not\models o_1 \succ o_2$. An *improving flip* is the flip of the value of a single variable X_k within an outcome to directly compute a preferred (better) outcome based on $CPT(X_k)$. *Note that the preference relation induced from a CP-net is generally not complete [2].* Two outcomes o_1 and o_2 may also be incomparable according to $\mathcal{N}$ (written as $\mathcal{N} \models o_1 \bowtie o_2$). $\mathcal{N} \models o_1 \bowtie o_2$ iff $\mathcal{N} \not\models o_1 \succ o_2$ and $\mathcal{N} \not\models o_2 \succ o_1$.

3 Reasoning with CP-Net Topologically

3.1 The Topological Order of Variables

During the preference elicitation process, the agent specifies its preference following a particular order of variables: the context of parent features must be specified before giving out the total order over the domain of a variable. Given an acyclic CP-net $\mathcal{N}$ over a set of n variables $\mathbf{V} = \{X_1, \ldots, X_n\}$, a topological order of variables $\sigma = X_1 > \cdots > X_n$ is a linear order on the variables from ancestors to descendants that is consistent with $\mathcal{N}$, i.e. for each parent variable X_k of a variable X_ℓ, we have $X_k > X_\ell$. Please note that for an acyclic CP-net $\mathcal{N}$, there may exist more than one topological order of variables. For instance, both the order of $X_1 > X_2 > X_3$ and $X_2 > X_1 > X_3$ are consistent with the structure of the example CP-net in Figure 1(a).

Definition 1. *Given an acyclic CP-net $\mathcal{N}$ over $\mathbf{V} = \{X_1, \ldots, X_n\}$, for any linear order $\sigma = X_1^\sigma > \cdots > X_n^\sigma$ over $\mathbf{V}$ that is consistent with $\mathcal{N}$, we define $\mathbf{Z}^\sigma = \{X_1^\sigma, \ldots, X_n^\sigma\}$ to be an ordered set of varaibles corresponding to σ; and $\mathbf{Z}_k^\sigma = \{X_1^\sigma, \ldots, X_k^\sigma\}$ $(1 \leq k \leq n)$ to be the k^{th} prefix subset of $\mathbf{Z}^\sigma$, which consists of the first k variables in $\mathbf{Z}^\sigma$.*

Restricting to the k^{th} prefix subset of variables $\mathbf{Z}_k$, we define that *ceteris paribus* preference between the values to a prefix subset $\mathbf{Z}_k$ as follows:

Definition 2. *Given an acyclic CP-net $\mathcal{N}$, let σ be a topological order that is consistent with $\mathcal{N}$, and $\mathbf{Z}^\sigma = \{X_1^\sigma, \ldots, X_n^\sigma\}$ be the ordered set of variables corresponding to σ. For any pair of prefix assignments $\mathbf{z}_k^\sigma$, $\mathbf{z}_k^{\sigma\prime}$ to the prefix subset of variables $\mathbf{Z}_k^\sigma$ ($\mathbf{z}_k^\sigma$, $\mathbf{z}_k^{\sigma\prime} \in D(\mathbf{Z}_k^\sigma)$), we say that $\mathbf{z}_k^\sigma$ is preferred to $\mathbf{z}_k^{\sigma\prime}$ certeris paribus, denoted by $\mathbf{z}_k^\sigma \overset{cp}{\succ} \mathbf{z}_k^{\sigma\prime}$ if and only if there exists an improving flipping sequence from $\mathbf{z}_k^{\sigma\prime}$ to $\mathbf{z}_k^\sigma$ resticting on the set of variables $\mathbf{Z}_k^\sigma$. Otherwise, $\mathbf{z}_k^\sigma \overset{cp}{\not\succ} \mathbf{z}_k^{\sigma\prime}$.*

Lemma 1. *Given an acyclic CP-net $\mathcal{N}$, let σ be a topological order that is consistent with $\mathcal{N}$, and $\mathbf{Z}^\sigma = \{X_1^\sigma, \ldots, X_n^\sigma\}$ be the ordered set of variables corresponding to σ. For any pair of prefix assignments $\mathbf{z}_k^\sigma$, $\mathbf{z}_k^{\sigma\prime} \in D(\mathbf{Z}_k^\sigma)$, let $\mathbf{W} = \mathbf{Z}^\sigma - \mathbf{Z}_k^\sigma$:*

- *If $\mathbf{z}_k^\sigma \overset{cp}{\succ} \mathbf{z}_k^{\sigma\prime}$, then for all $\mathbf{w} \in D(\mathbf{W})$, $\mathcal{N} \models \mathbf{z}_k^\sigma \mathbf{w} \succ \mathbf{z}_k^{\sigma\prime} \mathbf{w}$;*
- *else, for any $\mathbf{w}, \mathbf{w}' \in D(\mathbf{W})$, $\mathcal{N} \not\models \mathbf{z}_k^\sigma \mathbf{w} \succ \mathbf{z}_k^{\sigma\prime} \mathbf{w}'$.*

Proof. If $\mathbf{z}_k^\sigma \overset{cp}{\succ} \mathbf{z}_k^{\sigma\prime}$, then there exist an improving flipping sequence from $\mathbf{z}_k^{\sigma\prime}$ to $\mathbf{z}_k^\sigma$ restricting on the set of variables $\mathbf{Z}_k^\sigma$. Consequently, for all $\mathbf{w} \in D(\mathbf{W})$, there must also exist an improving flipping sequence from $\mathbf{z}_k^{\sigma\prime}\mathbf{w}$ to $\mathbf{z}_k^\sigma \mathbf{w}$ in the induced preference graph over the outcome space, which only flips the values of the set of variables $\mathbf{Z}_k^\sigma$ but keep the set of variables $\mathbf{W}$ unchanged. Thus, $N \models \mathbf{z}_k^\sigma \mathbf{w} \succ \mathbf{z}_k^{\sigma\prime} \mathbf{w}$. On the other hand, if $\mathbf{z}_k^\sigma \overset{cp}{\not\succ} \mathbf{z}_k^{\sigma\prime}$, there does not exist any improving flipping sequence from $\mathbf{z}_k^{\sigma\prime}$ to $\mathbf{z}_k^\sigma$ restricting on the set of variables $\mathbf{Z}_k^\sigma$. As each variable X in $\mathbf{Z}_k^\sigma$ is preferentially independent with the variables in $\mathbf{W}$ given the values to the set variables $\mathbf{Z}_k^\sigma - \{X\}$, for any $\mathbf{w}, \mathbf{w}' \in D(\mathbf{W})$, there must not exist any improving sequence from outcome $\mathbf{z}_k^{\sigma\prime}\mathbf{w}'$ to $\mathbf{z}_k^\sigma \mathbf{w}$. Consequently, $N \not\models \mathbf{z}_k^\sigma \mathbf{w} \succ \mathbf{z}_k^{\sigma\prime} \mathbf{w}'$. □

3.2 Generating the Preference Order over the Outcome Space Using RA-Tree

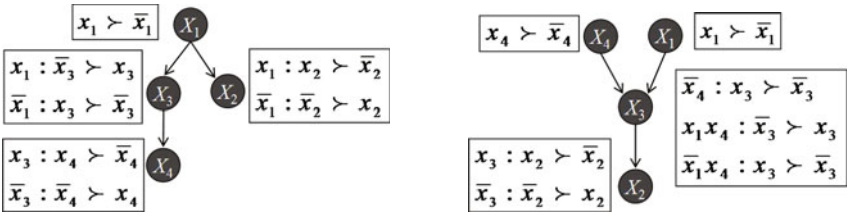
In this section, we conceptualize the assignment of the variables values as a tree, Relational Assignment Tree (called RA-Tree in the following sections), which constructs the agent's preference relations over the outcomes space following the topological order of variables. Given an acyclic CP-net $\mathcal{N}$ over a set of n variables(not necessary binary-valued), let σ be a topological order that is consistent with $\mathcal{N}$, and $\mathbf{Z}^{\sigma} = \{X_1^{\sigma}, \ldots, X_n^{\sigma}\}$ be the ordered set of variables corresponding to σ. A RA-Tree over $\mathbf{Z}^{\sigma}$ is a M-ary tree (M denotes the maximum domain size of the variables) that:

1. The depth of the tree is n; the root node being at level 0 represents an empty assignment to the variables and then each level k represents the value assignments to the prefix subset $\mathbf{Z}_k^{\sigma}$.
2. At each level k, we extends each node $\mathbf{z}_{k-1}^{\sigma}$ from the upper $(k-1)$ level with the values assign to the next variable $X_k^{\sigma} \in \mathbf{Z}^{\sigma}$; and the values of X_k^{σ} are ordered preferentially decreasingly from left to right *w.r.t.* $CPT\left(X_k^{\sigma}\right)$, given $\mathbf{z}_{k-1}^{\sigma}$;
3. At each level k, there is an **edge** from a node $\mathbf{z}_k^{\sigma\prime}$ to a node $\mathbf{z}_k^{\sigma}$ ($\mathbf{z}_k^{\sigma}$ and $\mathbf{z}_k^{\sigma\prime}$ differ on a single variable) if and only if:
   - $\mathbf{z}_k^{\sigma}$ and $\mathbf{z}_k^{\sigma\prime}$ have the same parent node $\mathbf{z}_{k-1}^{\sigma}$, and given $\mathbf{z}_{k-1}^{\sigma}$, the value assigned to variable X_k^{σ} by $\mathbf{z}_k^{\sigma}$ is preferred to that assigned by $\mathbf{z}_k^{\sigma\prime}$, i.e. $x \succ^{X_k^{\sigma}|\mathbf{z}_{k-1}^{\sigma}} x'$ where $x = \mathbf{z}_k^{\sigma}[X_k^{\sigma}]$ and $x' = \mathbf{z}_k^{\sigma\prime}[X_k^{\sigma}]$;
   - or, there is an edge from node $\mathbf{z}_{k-1}^{\sigma}{}'$ to node $\mathbf{z}_{k-1}^{\sigma}$ ($\mathbf{z}_{k-1}^{\sigma}$ and $\mathbf{z}_{k-1}^{\sigma}{}'$ are the parent nodes of $\mathbf{z}_k^{\sigma}$ and $\mathbf{z}_k^{\sigma\prime}$ respectively), $\mathbf{z}_k^{\sigma}[X_k^{\sigma}] = \mathbf{z}_k^{\sigma\prime}[X_k^{\sigma}] = x$, and given $\mathbf{z}_{k-1}^{\sigma}$ and $\mathbf{z}_{k-1}^{\sigma}{}'$, there does not exist $x'' \in D\left(X_k^{\sigma}\right)$ and $x'' \neq x$, such that it satisfies: $x \succ^{X_k^{\sigma}|\mathbf{z}_{k-1}^{\sigma}} x''$ and $x'' \succ^{X_k^{\sigma}|\mathbf{z}_{k-1}^{\sigma}{}'} x$.

At each level of the RA-Tree, from left to right is a non-increasing order over the prefix assignments at that level. Based on the transitive closure of the nodes at each level k, $\mathbf{z}_k^{\sigma} \overset{cp}{\succ} \mathbf{z}_k^{\sigma\prime}$ if and only if there is an edge or directed path form $\mathbf{z}_k^{\sigma\prime}$ to $\mathbf{z}_k^{\sigma}$. Moreover, each leaf node corresponds to a complete assignment to all the variables (i.e. outcomes), and the transitive closure at the lowest level of the RA-Tree specifies the (asymmetric) partial order over the outcome space induced by the CP-net $\mathcal{N}$ (see Figure 1(b) and Figure 1(c) the RA-Tree and the corresponding preference order over the outcome space of the example CP-net in Figure 1(a)).

4 Making Social Choices with RA-Trees

In many situations, we need to represent and reason about the simultaneous preferences of several agents, and to aggregate such preferences [9,8]. Given a set of CP-nets of the agents, our goal for collective decision-making should be to determine either one, some, or all outcomes that are not dominated by any other outcomes corresponding to the given aggregation rule [9]. Based on the assumption that the agents' CP-nets are acyclic and do not model indifference, we define the following definition of dominance in collective decision-making with multiple agents' CP-nets:



(a) CP-net of Agent 1 (b) CP-net of Agent 2

Fig. 2. Two examples of CP-nets

Definition 3. *Given a collection of the agents CP-nets* $\boldsymbol{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_m\}$ *and two alternatives* o, o', *we say that* o *dominates* o' *if and only if for every agent,* o' *is not preferred to* o *and there is at least one agent who prefers* o *to* o'*:* $\forall \mathcal{N}_i \in \boldsymbol{N}$, $\mathcal{N}_i \not\models o' \succ o$ *and* $\exists \mathcal{N}_j \in \boldsymbol{N}$, $\mathcal{N}_j \models o \succ o'$. *An outcome* o^* *is optimal if and only if it is not dominated by any other outcomes.*

Given a collection of CP-nets $\mathbf{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_m\}$ over a set of variables $\mathbf{V} = \{X_1, \ldots, X_n\}$, we first define a topological order of variables for each agent i's CP-net $\mathcal{N}_i$: $\sigma_i = X_1^{\sigma_i} > \cdots > X_n^{\sigma_i}$. To be more efficient, we try to prune out the dominated prefix assignments from the higher levels. Thus, the topological order of agents' CP-nets should be defined as similar as possible. In this paper, we define the following rules for choosing the topological order for the agents CP-nets.
Let $\mathcal{G}_i$ denote the relation graph over the set of domain variables of an agent i's CP-net $\mathcal{N}_i$. For any pair of variables X_k, X_ℓ:

- if, for all $i \in \{1, \ldots, m\}$, there is a directed path from X_k to X_ℓ in $\mathcal{G}_i$, then we have $X_k > X_\ell$ in σ_i;
- if, for all $i \in \{1, \ldots, m\}$, there is no directed path from X_ℓ to X_k in $\mathcal{G}_i$ and $\exists j \in \{1, \ldots, m\}$, there is a directed path from X_k to X_ℓ in σ_j. Then for all $i \in \{1, \ldots, m\}$, we have $X_k > X_\ell$ in σ_i.
- if, for all $i \in \{1, \ldots, m\}$, there is no directed path between X_k and X_ℓ, we then define a common order that every agent follows, e.g. for all $i \in \{1, \ldots, m\}$, we have $X_k > X_\ell$ in σ_i.

Example 1. Consider the example CP-nets of 2 agents over a set of 4 variables in Figure 2. For agent 1's CP-nets, according to the relation graph, we have the following strict ordering over the variables: $X_1 > X_2$, $X_1 > X_3$, $X_1 > X_4$ and $X_3 > X_4$. There is no directed path between X_2 and X_4 (resp. X_2 and X_3), however, according to agent 2's CP-net, there is a directed path from X_4 to X_2 (resp. from X_3 to X_2). Consequently, the topological order chosen for agent 1's CP-net is $\sigma_1 = X_1 > X_3 > X_4 > X_2$. Similarly, the topological order chosen for agent 2 is $\sigma_2 = X_1 > X_4 > X_3 > X_2$.

After the topological order of each agent's CP-net is determined, we can execute the RA-optimal procedure. RA-optimal procedure involves a $n-$iteration (n is the number of variables) process of pruning dominated prefix assignments. At each iteration ($k \leq n$), we traverse each agent's RA-Tree at a specific level and aggregate multiple

agents' preference over the prefix assignments. Please note that we don't assume that we have the complete RA-Tree of each agent in hand, but construct a partial RA-Tree for the remaining node assignments in the RA-Tree. For each level ℓ ($1 < \ell \leq n$), we only construct the children nodes of the remaining nodes at level $\ell - 1$. Given a collection of CP-nets $\mathbf{N} = \{\mathcal{N}_1, \ldots, \mathcal{N}_m\}$, let $\sigma_i = X_1^i > \cdots > X_n^i$ ($i \in \{1, \ldots, m\}$) be the chosen topological order of $\mathcal{N}_i$, $\mathbf{Z}^{\sigma_i} = \{X_1^{\sigma_i}, \ldots, X_n^{\sigma_i}\}$ be the ordered set of variables corresponding to σ_i and T_i be the partial RA-Tree of agent i corresponding to σ_i:
For $k = 1$ **To** n

Step1: For each agent i's CP-net $\mathcal{N}_i$, consider the k^{th} variable and its parents in all other agents' CP-nets. Let $X = X_k^{\sigma_j}$ be the k^{th} variable in $\mathbf{Z}^{\sigma_j}$ of agent j ($j \in \{1, \ldots m\}$ and $j \neq i$) and $\mathbf{S}_j = \{X\} \cup Pa_j(X)$. Let $\mathbf{S} = \bigcup_{j \in \{1,\ldots,m\}, j \neq i} \mathbf{S}_j$. Suppose that $\mathbf{Z}_\ell^{\sigma_i}$ is the smallest prefix subset of variables of agent i such that $\mathbf{Z}_\ell^{\sigma_i}$ contains all the variables in $\mathbf{S}$ ($\mathbf{S} \subseteq \mathbf{Z}_\ell^{\sigma_i}$). We traverse the ℓ^{th} level of Agent i's RA-Tree T_i.[2] For the simplicity of explanation, we present the proposed procedure under two-party setting; however, the proposed approach is feasible in multi-agent case according to the same principle. For all $x, x' \in D(X)$, we consider $CPT_j(X)$ of agent j and the ℓ^{th} level of agent i's RA-Tree:

- if for Agent j, x is unconditionally preferred to x' *w.r.t* $CPT_j(X)$, then delete all the prefix assignments $\mathbf{z}_\ell^{\sigma_i\prime}$ at level ℓ of T_i for which: $\mathbf{z}_\ell^{\sigma_i\prime}[X] = x'$, and there is an edge or directed path from $\mathbf{z}_\ell^{\sigma_i\prime}$ to any of the remaining prefix assignments $\mathbf{z}_\ell^{\sigma_i}$ that assigns x to variables X ($\mathbf{z}_\ell^{\sigma_i}[X] = x$). All children nodes of the deleted nodes at the lower levels should also be deleted.
- or, for any remaining prefix assignments $\mathbf{z}_\ell^{\sigma_i}, \mathbf{z}_\ell^{\sigma_i\prime} \in D(\mathbf{Z}_\ell^{\sigma_i})$ that assign the same value $\mathbf{u}$ to $Pa_j(X)$, but a different value to variable X: $\mathbf{z}_\ell^{\sigma_i}[X] = x$ and $\mathbf{z}_\ell^{\sigma_i\prime}[X] = x'$, if for Agent j, given the parent context $\mathbf{u}$, x is conditionally preferred to x', then delete all the prefix assignments $\mathbf{z}_\ell^{\sigma_i\prime}$ at the ℓ^{th} level of T_i if there is an edge or directed path from $\mathbf{z}_\ell^{\sigma_i\prime}$ to $\mathbf{z}_\ell^{\sigma_i}$. All children nodes of the deleted nodes at the lower levels should also be deleted.

Step2: In this step, we need to delete the infeasible prefix assignments in the agents' RA-Trees, as the remaining assignments in one agent's RA-Tree may have been deleted when traversing the other agent's RA-Trees. For instance, if the assignment x_1x_2 to variables X_1 and X_2 is deleted when traversing T_1, then all the lower level prefix assignments are deleted in T_1. If the assignment $x_1x_2x_3$ to variables X_1, X_2 and X_3 is remaining in T_2, then we delete the infeasible assignment $x_1x_2x_3$ in T_2. Moreover, when $\ell = k$, ($\mathbf{Z}_\ell^{\sigma_i} = \mathbf{Z}_k^{\sigma_j}$), we need to check if there exist some pairs of prefix assignments $\mathbf{z}_k$, $\mathbf{z}_k{}'$ at level k (ℓ), such that there is no path between $\mathbf{z}_k$ and $\mathbf{z}_k{}'$ in T_i (resp. T_j) but there is a path from $\mathbf{z}_k{}'$ to $\mathbf{z}_k$ in T_j (resp. T_i), then we delete $\mathbf{z}_k{}'$ in both T_i and T_j.

At the end of the RA-Optimal procedure, we obtain a set of optimal outcomes (the remaining leaf nodes in the agents' partial RA-Tree), which is the basis for choosing

[2] Please note that if only the first ℓ'^{th}($\ell' < \ell$) level of T_i has been constructed, then we will need to construct T_i from the remaining nodes of the ℓ'^{th} level to the ℓ^{th} level.

the final outcome preferred by both agents. As to proof the optimality of the proposed approach, we first extend the following corollary for ordering queries[3] from [2].

Corollary 1. *Let $\mathcal{N}_i$ be an acyclic CP-net of agent A_i, and o, o' be a pair of outcomes. If there exists a variable X in $\mathcal{N}_i$, such that*

- *X is a root variable in $\mathcal{N}_i$, o assigns a more preferred value to X than that assigned by o': $x \succ^{X}_{\mathcal{N}_i} x'$, where $x = o[X]$ and $x' = o'[X]$;*
- *Or, o and o' assign the same values to all ancestors of X in $\mathcal{N}_i$, and given the assignment provided by o (and o') to $Pa_i(X)$, o assigns a more preferred value to X than that assigned by o': $x \succ^{X|\mathbf{u}}_{\mathcal{N}_i} x'$, where $x = o[X]$, $x' = o'[X]$ and $\mathbf{u} = o[Pa_i(X)] (= o'[Pa_i(X)])$;*

then $\mathcal{N}_i \not\models o' \succ o$

Now, we prove that the proposed RA-Optimal procedure guarantees optimality.

Theorem 1. ***(completeness)*** *The outcomes that are deleted during the RA-Optimal procedure are not optimal.*

Proof. At each iteration, during Step 1, for any deleted prefix partial assignment $\mathbf{z}_\ell^{\sigma_i\prime}$, there must exist another prefix partial assignment $\mathbf{z}_\ell^{\sigma_i}$, such that: $\mathbf{z}_\ell^{\sigma_i} \overset{cp}{\succ}_{\mathcal{N}_i} \mathbf{z}_\ell^{\sigma_i\prime}$. Thus according to Lemma 1, $\forall \mathbf{w} \in D(\mathbf{Z}^{\sigma_i} - \mathbf{Z}_\ell^{\sigma_i})$, $N_i \models \mathbf{z}_\ell^{\sigma_i}\mathbf{w} \succ \mathbf{z}_\ell^{\sigma_i\prime}\mathbf{w}$. As $\mathbf{Z}_\ell^{\sigma_i}$ contains the k^{th} variable and its parents of agent j, according to the criteria of deleting prefix assignments in Step 1, there must exist another prefix assignment $\mathbf{z}_\ell^{\sigma_i}$, such that there must exist a variable X, $x \succ^{X}_{\mathcal{N}_j} x'$, or, $x \succ^{X|\mathbf{u}}_{\mathcal{N}_j} x'$ where $x = \mathbf{z}_\ell^{\sigma_i}[X]$, $x' = \mathbf{z}_\ell^{\sigma_i\prime}[X]$ and $\mathbf{u}$ is the parent context assigned by $\mathbf{z}_\ell^{\sigma_i}$ and $\mathbf{z}_\ell^{\sigma_i\prime}$. Thus, based on Corollary 1, $\forall \mathbf{w}, \mathbf{w}' \in D(\mathbf{Z}^{\sigma_i} - \mathbf{Z}_\ell^{\sigma_i})$, $N_j \not\models \mathbf{z}_\ell^{\sigma_i\prime}\mathbf{w}' \succ \mathbf{z}_\ell^{\sigma_i}\mathbf{w}$. Thus all the outcomes with the deleted prefix assignment $\mathbf{z}_\ell^{\sigma_i\prime}$ is not optimal. Moreover, the deleted prefix assignments in Step 2 and the corresponding outcomes are also not optimal according to Definition 3. □

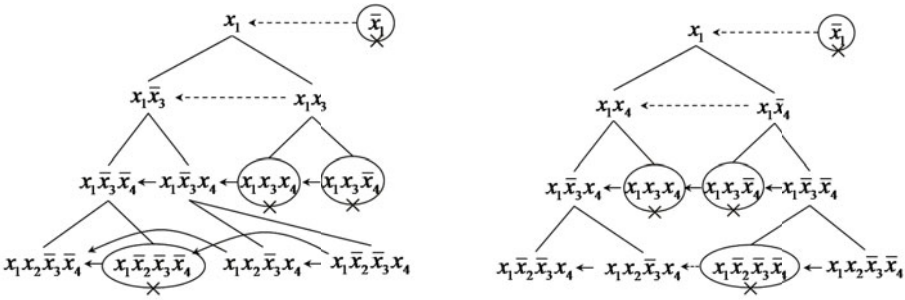
Theorem 2. ***(soundness)*** *The outcomes generated by RA-Optimal procedure are optimal.*

Proof. At each iteration, we guarantee that all the dominated prefix assignments at the current level are deleted during Step 1 and Step 2. When we finish the n^{th} iteration, all the remaining leaf nodes are not dominated by any other leaf nodes. Thus the outcomes (remaining leaf nodes) generated by RA-Optimal procedure are optimal. □

Example 1 (*continued*). Now, we demonstrate the execution of the proposed approach with our running example shown in Figure 2. We choose the topological ordered set $\mathbf{Z}^{\sigma_1} = \{X_1, X_3, X_4, X_2\}$ for Agent 1 and $\mathbf{Z}^{\sigma_2} = \{X_1, X_4, X_3, X_2\}$ for Agent 2.

In the 1^{st} iteration, as $X_1^{\sigma_2} = X_1$ and $Pa_2(X_1) = \emptyset$, $\mathbf{S} = \{X_1\} \cup Pa_2(X_1) = \{X_1\}$, $\mathbf{Z}_1^{\sigma_1}$ is the smallest prefix subset of Agent 1 that contains all the variables in $\mathbf{S}$. We construct the first level of T_1, as Agent 2 prefer x_1 to $\bar{x}_1$ *w.r.t* $CPT_2(X_1)$ and there

[3] An ordering query, given a CP-net $\mathcal{N}_i$ of an agent A_i and a pair of outcomes o and o', asks whether $\mathcal{N}_i \not\models o' \succ o$.



(a) Partial RA-Tree of Agent 1 (b) Partial RA-Tree of Agent 2

Fig. 3. Searching for optimal outcome set

(a) Agent 1 (b) Agent 2

Fig. 4. Preferences over the set of optimal outcomes

is a directed edge from $\bar{x}_1$ to x_1 in T_1, we delete $\bar{x}_1$ in T_1 (see Figure 3(a)). Similarly, we also delete $\bar{x}_1$ in T_2. (see Fig 3(b)). The outcome space has been cut in half after the 1^{st} iteration as all other outcomes with $\bar{x}_1$ will not be considered.

In the 2^{nd} iteration, for Agent 1, consider the second variable in Agent 2's topological order $X_2^{\sigma_2} = X_4$ and $Pa_2(X_4) = \emptyset$, thus $\mathbf{S} = \{X_4\} \cup Pa_2(X_4) = \{X_4\}$. We construct the 3^{rd} level (the smallest level containing $\mathbf{S}$) of T_1 from the remaining node $\mathbf{z}_1^{\sigma_1} = x_1$. Consider $CPT_2(X_4)$ of Agent 2, x_4 is unconditionally preferred to $\bar{x}_4$, thus $x_1x_3\bar{x}_4$ is deleted in T_1 as there is a directed edge from $x_1x_3\bar{x}_4$ to $x_1x_3x_4$. Similarly, as $X_2^{\sigma_1} = X_3$ and $Pa_1(X_3) = X_1$ we need to construct the 3^{rd} level of T_2 and two prefix assignments $x_1x_3x_4$ and $x_1x_3\bar{x}_4$ are deleted in T_2. Because given the same parent context x_1, Agent 1 preferred $X_3 = \bar{x}_3$ to $X_3 = x_3$, and there is a path from $x_1x_3x_4$ (resp. $x_1x_3\bar{x}_4$) to $x_1\bar{x}_3x_4$ in T_2. During Step 2, we deleted the infeasible prefix assignment $x_1x_3x_4$ in T_1. After the 2^{nd} iteration, only 2 prefix assignments over 3 variables X_1, X_3 and X_4 are remain.

In the 3^{rd} iteration, $X_3^{\sigma_2} = X_3$, $Pa_2(X_3) = \{X_1, X_4\}$ and $\mathbf{S} = \{X_3\} \cup Pa_2(X_3) = \{X_1, X_3, X_4\}$. Thus we traverse the 3^{rd} level of T_1 and similarly, we traverse 3^{rd} level of T_2. No prefix assignment is deleted in this iteration.

During the last iteration, we construct the 4^{th} level of T_1 (resp. T_2) with the remaining parent nodes $x_1\bar{x}_3\bar{x}_4$ and $x_1\bar{x}_3x_4$. No assignment is deleted during Step 1. In Step 2, as there is no directed path between $x_1\bar{x}_2\bar{x}_3\bar{x}_4$ and $x_1x_2\bar{x}_3x_4$ in T_1 and there is a directed path from $x_1\bar{x}_2\bar{x}_3\bar{x}_4$ to $x_1x_2\bar{x}_3x_4$ in T_2, thus $x_1\bar{x}_2\bar{x}_3\bar{x}_4$ is deleted in both T_1 and T_2. The RA-optimal procedure then ends and returns a set of three optimal outcomes $x_1x_2\bar{x}_3\bar{x}_4$, $x_1x_2\bar{x}_3x_4$ and $x_1\bar{x}_2\bar{x}_3x_4$. For any pair of outcomes in this set, Agent 1 and Agent 2 have conflict preferences over them (See Fig 4).

5 Conclusion and Future Work

In this paper, we have introduced RA-Tree, a new method to reason with CP-net topologically and generate the preference order over the outcome space for an individual agent. When more than one agent is involved, we have also provided an efficient procedure to find the optimal outcome set for multiple agents. The proposed procedure reduces the size of the outcome space and thus increases the search efficiency.

In this work, we have assumed so far that we have the complete information about the agents' CP-nets. However, in many applications, decision making is more distributed and agents' preferences are not common knowledge. Therefore, future work includes the exploration of possible ways for making social choices with CP-nets under incomplete information setting.

Acknowledgements

We thank anonymous reviewers for helpful comments. This work is partially supported by the ARC Discovery Grant DP0987380.

References

1. Arrow, K.J.: Social choice and individual values / by Kenneth J. Arrow. Wiley, New York (1951)
2. Boutilier, C., Brafman, R.I., Hoos, H.H., Poole, D.: CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. Journal of Artificial Intelligence Research 21, 135–191 (2004)
3. Boutilier, C., Brafman, R.I., Hoos, H.H., Poole, D.: Preference-based constrained optimization with CP-nets. Computational Intelligence 20, 137–157 (2004)
4. Goldsmith, J., Lang, J., Truszczynski, M., Wilson, N.: The computational complexity of dominance and consistency in CP-nets. J. Artif. Int. Res. 33(1), 403–432 (2008)
5. Lang, J., Xia, L.: Sequential composition of voting rules in multi-issue domains. Mathematical Social Sciences 57(3), 304–324 (2009)
6. Li, M., Vo, Q.B., Kowalczyk, R.: An efficient majority-rule-based approach for collective decision making with CP-nets. In: KR (2010)
7. Li, M., Vo, Q.B., Kowalczyk, R.: An efficient procedure for collective decision-making with CP-nets. In: ECAI, pp. 375–380 (2010)
8. Purrington, K., Durfee, E.H.: Making social choices from individuals' CP-nets. In: AAMAS 2007: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1–3. ACM, New York (2007)
9. Rossi, F., Venable, B., Walsh, T.: mCP nets: representing and reasoning with preferences of multiple agents. In: AAAI 2004: Proceedings of the 19th Conference on Artificial Intelligence, San Jose, California, USA, pp. 729–734 (2004)

Real-Time Coordination of Concurrent Multiple Bilateral Negotiations under Time Constraints

Khalid Mansour, Ryszard Kowalczyk, and Bao Quoc Vo

Center for Complex Software Systems and Services,
Swinburne University of Technology, Melbourne, Australia
{mwmansour,RKowalczyk,BVo}@groupwise.swin.edu.au

Abstract. This paper addresses a coordination problem during concurrent multiple bilateral negotiations. Real time decisions need to be taken during negotiation to ensure that the overall negotiation process is successful in terms of achieving a valuable agreement (an agreement with high utility). Existing approaches for managing concurrent multiple bilateral negotiation typically rely on historic information such as the probability distribution of the opponents' reservation prices, deadlines, or the results of the previous negotiations. This paper presents a novel heuristic coordination mechanism for coordinating concurrent multiple bilateral negotiations assuming no previous knowledge about the opponents. The proposed method uses two measures to evaluate the relative behavior of each opponent. The first measure uses the first-order differences of the current negotiation offers and calculates the relative concession for each opponent agent that depends on the mutual concessions of all agents involved in negotiation. The second measure uses a scoring function to evaluate each of the seller's last offer during negotiation. We use both measures to change the negotiation strategy during negotiation. The empirical results show that our model is more effective when compared with some existing approaches.

Keywords: agents, negotiation, coordination.

1 Introduction

When multiple agents work together for a common goal, their decisions need to be aligned in a coherent manner as the independent actions of the agents may not be successful in accomplishing their mission. For example, if two agents A and B are negotiating with their opponents, then agent A might not be able to accept an agreement before agent B does.

Negotiation is an effective decision making mechanism and dispute neutralization method that can be used by opponents having conflicting interests [14]. The negotiation is a process where two or more parties interact according to a certain protocol for the purpose of reaching an agreement over one or more issues. Price and reliability of a service are examples of negotiation issues. This paper presents a coordination problem from a buyer's perspective and focuses on

J. Li (Ed.): AI 2010, LNAI 6464, pp. 385–394, 2010.

the problem of managing concurrent multiple bilateral negotiations in an environment characterized by limited critical information about the opponents such as reservation values, deadlines and negotiation strategies.

The one-to-many form of negotiation is a process where one agent is negotiating with many other agents (> 1) for the purpose of reaching an agreement(s) [12]. The object of negotiation varies, agents may negotiate about buying or selling goods [13], procuring resources [1] and services [11] or negotiate about task assignment [15]. Regardless of the object of negotiation, the negotiating agent(s) seeks to reach a certain number of agreements. This work focuses on securing one agreement while negotiating with multiple opponents.

Most existing approaches of managing concurrent multiple bilateral negotiations typically assume a priori knowledge such as the probability distribution of the opponents' reservation prices [4], or historic data [12] about the previous negotiation encounters. We advance the state of the art by proposing a new coordination technique that uses the first-order differences of the agents' offers and a scoring function to evaluate the opponents' last offers in the current negotiation encounter in order to change the strategy of negotiation in terms of the amount of the concession in the next negotiation round.

The rest of the paper is organized as follows: section 2 discusses the related work, while section 3 introduces a formal description of the one-to-many negotiation form. Section 4 explains the coordination process and describes our coordination model, while section 5 describes the experimental settings and discusses the results of the experimental work. Finally, section 6 concludes the paper and points out some future research directions.

2 Related Work

During the last decade, work has been done to address the one-to-many form of negotiation as an alternative mechanism to the single-sided auction protocol [12] [6] [9] [7] [2] [1].

Adopting the one-to-many form of negotiation enables the buyer to propose counter offers to each individual seller using different negotiation strategies. Accordingly, the chance of reaching an agreement will be improved since each agent in the negotiation process can consider the previous offers aiming at predicting the preferences of its opponent and try to propose an offer that might improve the probability of reaching an agreement. For more details about the advantages of using the one-to-many negotiation form over the reverse English auctions, see [13] and [9].

The first explicit architecture for the one-to-many negotiation form was presented in [13] where the buyer agent consists of sub-negotiators and a coordinator. The authors [13] propose four different coordination strategies during concurrent multiple bilateral negotiation: desperate strategy in which the buyer agent accepts the first agreement and quits negotiations with all other sellers, patient strategy where the buyer agent makes temporary agreements with some

or all sellers during negotiation and holds on these agreements until all the remaining threads of negotiations are finished, then the buyer agent selects the agreement with the highest utility, optimized patient which is similar to the patient strategy except that it does not accept a new agreement with less utility than the highest existing one, and finally the manipulation strategies in which the coordinator changes the negotiation strategies of its sub-negotiators during negotiation which was left for future work. Our approach adopts the same idea as the manipulation strategies method where the agent changes its negotiation strategy during negotiation.

Our work is similar to some other existing work [12] [11] in terms of choosing the coordination approach that changes the strategy during negotiation. For example, the decision making technique in changing the negotiation strategies [12] during negotiation depends on historic information of previous negotiations in terms of agreements rate and utilities rate. The difference is that our approach does not rely on previous information and uses a different technique to change negotiation strategies.

While [9], [1] consider decommitment penalty during negotiation, we assume that the buyer agent incur no penalty for exercising decommitment during negotiation. Having granted the privilege only for the buyer agent to renege from a deal without a penalty, while forcing the seller agents to honour their agreements can be a realistic scenario in situations where the number of seller agents is large and/or the seller agents are offering infinite supply (e.g. information). In such cases, a seller agent might be satisfied to make deals with many potential buyers in a hope that some of these buyers will confirm their deals later.

Some heuristic methods were proposed to estimate the expected utility in both a synchronized multi-threaded negotiations and a dynamic multi-threaded negotiations [4]. The synchronized multi-threaded negotiations model considers the existing outside options for each single thread, while the dynamic multi-threaded negotiations considers also the uncertain outside options that might come in the future. In both cases, the methods assume a knowledge of the probability distribution of the reservation prices of the opponents. In many cases, this kind of information is not available.

While [2] proposes a decision making strategy using Markov chains to decide whether to accept the best available offer or to proceed in negotiation with a hope to achieve a better deal, our work assumes that the buyer can make temporary deals with his opponents as explained previously.

The AutONA (A System for Automated Multiple 1-1 Negotiation) [3] was tested in the HPs Experimental Economics Lab as an automation tool to reduce the operational procurement costs. Their experimental results show good performance. However, the AutONA model uses a probability distribution over prices per unit, parameterized by the properties that an option (a seller and a quantity) may have. However, the objective of negotiation in AutONA is to procure a number of units from many sellers, while in our model we assume that one agreement is required.

3 Formal Description

This section presents the one-to-many form of negotiation more formally. Figure 1 illustrates an architecture for the one-to-many negotiation form, in which d_i stands for a negotiation thread, s_i stands for a seller agent and each negotiation thread corresponds to one seller agent. The buyer agent can initiate a unique negotiation thread with each different seller. The architecture illustrated in Figure 1 is similar to the architecture proposed in [10] [11] [12] [9], and it will be the base for our formal description of the one-to-many negotiation form. The formal representation helps in logical grouping and management of the negotiation threads by the coordinator.

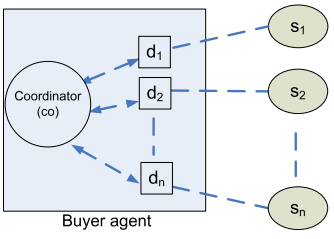


Fig. 1. One-To-Many Negotiation

We assume that a buyer agent b negotiates with a set of sellers $S = \{s_1, s_2, ..., s_n\}$. The following model presents a buyer's perspective. A seller's perspective can be constructed in a similar way. We assume the following:

1. The buyer agent b consists of a coordination unit co and a set of threads D. Formally, $b = \{co, D\}$,$D = \{d_1, d_2, ..., d_n\}$.
2. Given that $|D| = |S|$, let R_{ds} defines the relationship between the set of sellers S and the set of threads D, then $R_{ds} = \{(d_1, s_1), (d_2, s_2), ..., (d_n, s_n)\}$.
3. Agents negotiate about objects. An object $o_i \in O = \{o_1, o_2, ..., o_m\}$ represents a physical item, a logical item (e.g. electronic material), a resource, a task, or a service.
4. Each object o_i is represented by a set of issues I_i, $I_i = \{is_1^i, is_2^i, ..., is_k^i\}$.
5. Each thread d_i is associated with a set of parameters which represent the constraints of negotiation assigned to that particular thread and with an object o_i. Formally, $\forall\ d_i \in D,\ d_i$ is associated with a set $st_i = \{o_i,\ t_{max}^{d_i},\ w_i,\ y_i\}$, where $i \in \{1, 2, ..., |D|\}$, given that $t_{max}^{d_i}$, w_i, y_i stand for the deadline, the weight, and the negotiation tactic or strategy of a thread d_i respectively. The w_i reflects the relative importance of a certain object o_i assigned to a thread d_i.
6. Since each issue is_j can be represented by a set of constraints such as reservation value, quality level, weight, etc., then $\forall\ is_j \in I_i$, is_j can be represented by a set of constraints $cs_j = \{c_1^j, c_2^j, ..., c_r^j\}$.
7. Agents exchange offers associated with the elements of a set cs_j.

8. The power set of D (i.e. 2^D) contains all possible groupings of threads, given that each group is responsible for securing one agreement. The coordinator (co) selects a subset $G \subseteq 2^D \backslash \emptyset \; s.t \; \forall \; l_i, l_j \in G, (l_i \cap l_j = \emptyset \wedge \bigcup_{i=1}^{|G|} l_i = D)$.
9. Each $l_i \in G$ has a number of threads that is/are responsible for negotiating about a distinct object. For Example, If $D = \{d_1, d_2\}$, then $2^D \backslash \emptyset = \{\{d_1\}, \{d_2\}, \{d_1, d_2\}\}$. If $G = \{\{d_1, d_2\}\}$ then G has two threads that negotiate with two sellers for the purpose of buying the same object while one agreement is needed. If $G = \{\{d_1\}, \{d_2\}\}$, then the purpose is to secure two agreements for two distinct objects from two different providers. It can happen, for example, that o_1 is assigned to $l_1 = \{d_1\} \in G$ and o_2 is assigned to $l_2 = \{d_2\} \in G$ and at the same time both objects o_1 and o_2 are assigned the same issue set I_i. For example, a book and a pen are two distinct objects, but both can be represented by the same set of negotiation issues I_i (e.g. $I_i = \{price, quality\}$).
10. If a seller s_i offers m number of objects, and if the thread d_i is required to buy the m objects from the seller s_i, then the thread d_i would negotiate about $\bigcup_{j=1}^{m} \{I_j\}$ issues, where object o_j is associated with I_j.
11. If seller agents s_i and s_j offer more than one object (e.g. o_1 and o_2) and the buyer agent seeks to buy both o_1 and o_2 objects, then s_i should create two seller threads (i.e. $s_{i_1} \rightarrow o_1$ and $s_{i_2} \rightarrow o_2$) and s_j should also create two seller threads (i.e. $s_{j_1} \rightarrow o_1$ and $s_{j_2} \rightarrow o_2$) then the buyer agent would have four threads (i.e $bd = \{\{d_1, d_2\}, \{d_3, d_4\}\}$ and $|G| = 2$). The mapping between the agents would be as follows: $R_{ds} = \{(d_1, s_{i_1}), (d_2, s_{j_1}), (d_3, s_{i_2}), (d_4, s_{j_2})\}$. In this case, the buyer agent seeks two agreements, one from the set $\{d_1, d_2\}$ for buying o_1 and another one from the set $\{d_3, d_4\}$ for buying o_2.

The cardinality of the subset G (i.e. $|G|$) determines the number of agreements required by the buyer agent. For example, if $|G| = 1$ then the scenario describes a situation where the buyer agent aims to secure one agreement while negotiating with multiple sellers. Our coordination mechanism assumes that $|G| = 1$.

4 The Coordination Process

We present the coordination mechanism as a system that has an input, process and output. The independent variables of negotiation such as the reservation values and deadlines are part of the required input for the coordination mechanism. Opponent's responses in terms of their counter offers are important feedback during negotiation. Other information such as the possible arrivals of new outside options during negotiation can affect the coordination method. However, in our model, we only consider the feedback in terms of the opponents' counter offers, since in many cases, other information may not be available. The outputs of the coordination process determine the effectiveness of the process in reaching its goals. Various metrics can be used to measure the effectiveness of the process, such as the utility and agreement rate.

The idea of our coordination of concurrent negotiations (CCN) model is to measure the relative behavior of each seller agent during negotiation in terms of

its concessions. In addition, each opponent's last offer is evaluated using a scoring function. The two measures form a metric value that is used to change the strategy of each buyer's thread in terms of their concessions in the next negotiation round. We assume that the agents use the polynomial function proposed in [5] to determine the value of the next offer where $\alpha(t) = (t/t^a_{max})^{1/\beta}$, where a stands for an agent, t^a_{max} stands for the deadline of agent a, t is the current time while the β value determines the convexity degree of the concession curve.

The proposed model uses the first-order differences of each agent's offers during the current negotiation for the first measure, while it uses a scoring function $V(x^t_{s_i \to d_i}) = (max^b - x^t_{s_i \to d_i})/(max^b - min^b)$ for the second measure, where $x^t_{s_i \to d_i}$ stands for an offer from a seller s_i to a buyer's thread d_i at time t, min^b and max^b are the reservation intervals of the buyer.

Let $\Delta fc^t_{s_i}$ and $\Delta fc^t_{d_i}$ stand for the sequence of the first-order differences of a seller s_i offers and the sequence of the first-order differences of a thread's d_i offers at time t, respectively. Then $\Delta fc^t_{s_i} = (x^{t-h}_{s_i \to d_i} - x^t_{s_i \to d_i})$, $\Delta fc^t_{d_i} = (x^t_{d_i \to s_i} - x^{t-h}_{d_i \to s_i})$, where h stands for a number of steps back in the history of the current negotiation offers ($h <= t$). At time t and for each $(d_i, s_i) \in R_{ds}$, we calculate the difference between $\Delta fc^t_{s_i}$ and $\Delta fc^t_{d_i}$ as shown in Equation 1.

$$\forall (d_i, s_i) \in R_{ds}, \Delta fc^t_i = \Delta fc^t_{s_i} - \Delta fc^t_{d_i} \tag{1}$$

$fc^t = \{\Delta fc^t_1, \Delta fc^t_2, ..., \Delta fc^t_n\}$, where fc^t is a set representing the relative concession of each seller $s_i \in S$ at time t. To analyze how each seller agent's behavior is compared to the best negotiating seller(s) from the point of view of the buyer agent in terms of its relative concession, we normalize the values in the set fc^t, $\forall \Delta fc^t_i \in fc^t, \Delta fc^t_{i_{norm}} = (\Delta fc^t_i - Min(fc^t))/(Max(fc^t) - Min(fc^t))$.

A seller agent s_i having $\Delta fc^t_{i_{norm}} = 0$ indicates that s_i is the most unfavorite agent in terms of its relative concession up to time t, while $\Delta fc^t_{i_{norm}} = 1$ means that s_i is the most favorite seller agent in terms of the amount of its relative concession up to time t.

Computing the score value measure is done by calculating a scoring value for each seller's last offer using the function V and then normalizing the results. Each normalized value $v^t_{i_{norm}}$ corresponds to a seller agent s_i. Up to this point of time t, the coordinator unit co has an overall view of the relative behavior of each seller agent. The combined two measures is given by Equation 2.

$$\epsilon^t_i = (1 - \gamma)\Delta fc^t_{i_{norm}} + \gamma v^t_{i_{norm}} \tag{2}$$

given that $\gamma \in [0, 1]$ and $\epsilon^t_i \in [0, 1]$. At each negotiation round where $t > 1$ given that the first round starts at $t = 0$, ϵ^t_i value will be calculated for each seller then the coordinator (co) changes the strategy (concession rate or β^{t+1}_i value for the next negotiation round) for each thread $d_i \in D$ according to the following formula:

$$\beta^{t+1}_i = \begin{cases} c - c\epsilon^t_i, & \epsilon^t_i \in [0, 0.5) \\ Max[1 - \epsilon^t_i, \rho], & \epsilon^t_i \in [0.5, 1] \end{cases} \tag{3}$$

where c stands for the value of concession($c >> 1$). In our experiments, we use $c = 10$. The ρ value determines the minimum β^{t+1}_i value when $\epsilon^t_i = 1$. It is

obvious from Equation 3 that the strategy choice is characterized by exploitation of the situation. The *co* chooses to play tough with lenient opponents while playing lenient with tough opponents. This decision reflects a certain belief about the social behavior of the seller agents. If the coordinator has a different belief, then the strategy updating mechanism may change. A summary of the *CCN* algorithm is outlined in Figure 2.

1: **for all** $s_i \in S, d_i \in D$ **do**	9: NOMALIZE(v_i^t)
2: $\Delta fc_i^t = \Delta fc_{s_i}^t - \Delta fc_{d_i}^t$	10:**for all** $s_i \in S$
3: $fc^t[i] = \Delta fc_i^t$	11: $\epsilon_i^t = (1-\gamma) * \Delta fc_{i_{norm}}^t + \gamma * v_{i_{norm}}^t$
4: **end for**	12: **end for**
5: NORMALIZE(fc^t)	13: **if** $\epsilon_i^t \in [0, 0.5)$ **then** $\beta_i^{t+1} = c - c * \epsilon_i^t$
6: **for each** $s_i \in S$	14: **else**
7: $v_i^t = V(x_{s_i \to d_i}^t)$	15: $\beta_i^{t+1} = Max[1 - \epsilon_i^t, \rho]$
8: **end for**	16: **end if**

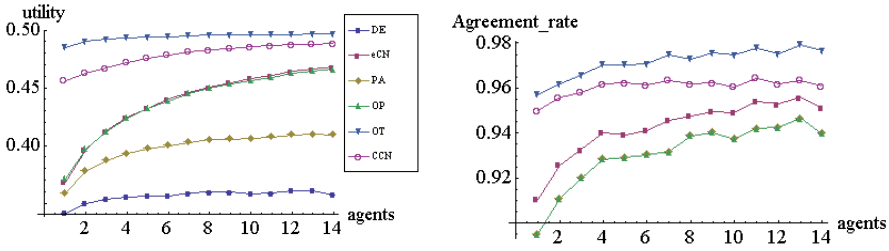
Fig. 2. The CCN algorithm

5 Empirical Evaluation

To validate the proposed coordination model, we build 75 (5 different overlap percentages x 3 different deadlines x 5 different γ values) different test cases to ensure that our coordination (CCN) mechanism works in a wide range of possible negotiation environments. For simplicity, we assume that all agents use the time dependent tactics proposed by [5]. In each experiment, all buyer's threads start having the same deadline, reservation values, and the same negotiation strategy in terms of their concession (i.e. β value). Only the negotiation strategies will be changed for the buyer's threads. Each seller agent selects a random negotiation strategy before the start of negotiation. The reservation values and deadlines of the sellers are selected randomly from the same distribution.

In this paper, we benchmark our strategy against five other strategies, namely, the desperate strategy (DE), the patient strategy (PA), the optimized patient strategy (OP) [13], the (eCN) strategy [12], and finally the optimal strategy (OT). The optimal strategy assumes full knowledge about the opponents with regard to their reservation values, deadlines and negotiation strategies. The eCN uses historic data from previous negotiations and changes the negotiation strategy of a buyer's thread after the corresponding opponent agent has been classified either as a conceder or non conceder. We assume that the agents negotiate about one issue (e.g. price). The feedback is represented by (ϵ_i^t) which is calculated as explained in Section 4. The metrics that we use to evaluate the effectiveness of the coordination process are the utility gain and the agreement rate.

For the experimental settings, we use various degrees of overlaps between the reservation intervals of the negotiating agents, namely, 100%, 75%, 50%, 25% and 5%. The deadline of the buyer agent has three states: equal, shorter or longer than the sellers' deadlines. We also use 5 classes of γ values, $\gamma \in \{0, 1, 0.5, (0.5, 1], [0.0, 0.5)\}$. The minimum interval values for the buyer agent



(a) Average Utility (b) Agreement Rate

Fig. 3. Average Utility subfigures (a) and Agreement Rate (b)

where selected from [0, 20] and maximum interval values where selected from the interval [30, 50]. We run 75 different experiments, one for each different negotiation setting and each experiment was iterated 100 times. Figure 3(a) shows the average utility rate and the average agreement rate for each type of strategy over the 75 different negotiation settings. The CCN method performs better than all other strategies except for the optimal one (OT). Figure 3(a) shows that our method outperforms the best control (other than the optimal) method by about 4.4% when the number of agents is 15 and up to 24% when the number of agents is 2. Since our results come from unknown distribution, we use the nonparametric Mann-Whitney U [8] test to compare between the utility results of the CCN method and the next second best (i.e. eCN) method, the result of the test shows that the difference between the two samples is highly significant ($P < 0.001$, two-tailed test). The reason is that our coordination strategy measures each opponent's behavior in relation to the most favorite opponent's behavior from the buyer's point of view which helps in selecting the appropriate strategy for each thread. As mentioned earlier, the exploitation of the current situation guides the sellection of strategies for the buyer's threads.

Figure 3(a) shows that as the number of agents increases, the strategies CCN, eCN and OP approach the maximum utility which can be justified by the fact that when the number of agents increases, the probably of getting an agreement with higher utility increases given that these strategies does not accept a new agreement with a less utility than the maximum existing one.

The three controls, DE, PA and OP have the same agreement rate in all different negotiation environments due to the fact that they keep negotiating until they reach an agreement, if possible. Figure 3(b) shows that the CCN records the highest number of agreements except for the optimal (OT). The reason is that the CCN method monitors the relative behavior of the seller agents and changes the threads' negotiation strategies accordingly. In addition, the CCN monitors the number of outside options (i.e., opponents) left in negotiation, and if the number of the outside options for the last negotiating buyer's thread is zero, then the coordinator accepts a deal from the last remaining opponent if its offer is less than or equal to the reservation value of the buyer agent.

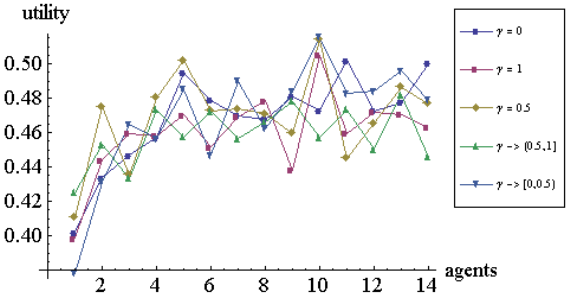


Fig. 4. Different γ values

To study the difference between the two evaluation measures (i.e. the relative concession and scoring value) in terms of their efficiency, we run five different experiments using the CCN method and each time we iterate the experiment 500 times. Each different experiment has a certain value for γ which reflects the weight of each measure in determining the ϵ_i^t value. The five different values of γ are $(0, 1, 0.5, (0.5, 1], [0.0, 0.5))$. The results of the experiments (see Figure 4) show that when $\gamma \in [0.5, 1]$, the utility values seems low when compared to the other γ values. Again we apply the Mann-Whitney U test for the utility gain when $\gamma \in [0.5, 1]$ and the utility gain when $\gamma = 0.5$, because the graph show a higher utility values when $\gamma = 0.5$. The result of the Mann-Whitney U test shows that the two samples are not significantly different ($P >= 0.05$, two-tailed test). We conclude that both measures have similar effect on utility gain.

6 Conclusions and Future Work

In the one-to-many form of negotiation, we propose a novel heuristic method for securing one valuable agreement. The decision making mechanism for selecting the next concession value in our model depends on analyzing the relative behavior of each opponent during current negotiation. We only consider the opponents' offers during current negotiation to decide on a new negotiation strategy. The empirical results show that our coordination mechanism achieves better results in terms of utility gain and number of agreements when compared with some exiting methods. In addition, further experiments prove that both the concession measure and the scoring value measure are similar in their efficiency. Furthermore, this paper presents a formal description for the one-to-many negotiation model. In future work we intend to investigate more complex situations where the objective of the buyer agent is to secure more than one agreement (i.e. $|G| > 1$).

References

1. An, B., Lesser, V., Irwin, D., Zink, M.: Automated Negotiation with Decommitment for Dynamic Resource Allocation in Cloud Computing. In: 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, pp. 981–988 (2010)
2. An, B., Sim, K.M., Miao, C.Y., Shen, Z.Q.: Decision making of negotiation agents using markov chains. Multiagent and Grid Systems 4, 5–23 (2008)
3. Byde, A., Yearworth, M., Bartolini, C.: AutONA: a system for automated multiple 1-1 negotiation. In: IEEE International Conference on E-Commerce, CEC 2003, pp. 59–67 (2003)
4. Cuihong, L., Giampapa, J., Sycara, K.: Bilateral negotiation decisions with uncertain dynamic outside options. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews) 36(1), 31–44 (2006)
5. Faratin, P.: Automated service negotiation between autonomous computational agents. PhD thesis, University of London (2000)
6. Gerding, E.H., Somefun, D.J.a., La Poutré, J.a.: Automated bilateral bargaining about multiple attributes in a one-to-many setting. In: Rauterberg, M. (ed.) ICEC 2004. LNCS, vol. 3166, p. 105. Springer, Heidelberg (2004)
7. Jun, Y., Ryszard, K., Jian, L., Chhetri, M.B., Keong, G.S., Jianying, Z.: Autonomous service level agreement negotiation for service composition provision. Future Gener. Comput. Syst. 23(6), 748–759 (2007)
8. Mann, H., Whitney, D.: On A Test of Whether One of Two Random Variables is Stochastically Larger Than the Other. Annals of Mathematical Statistics 18, 50–60 (1947)
9. Nguyen, T., Jennings, N.: Managing commitments in multiple concurrent negotiations. Electronic Commerce Research and Applications 4(4), 362–376 (2005)
10. Nguyen, T.D., Jennings, N.R.: A heuristic model for concurrent bi-lateral negotiations in incomplete information settings. In: 18th Int. Joint Conf. on AI, Acapulco, Mexico, pp. 1467–1469 (2003)
11. Nguyen, T.D., Jennings, N.R.: Concurrent bi-lateral negotiation in agent systems. In: Proceedings of the Fourth DEXA Workshop on E-Negotiations (2003)
12. Nguyen, T.D., Jennings, N.R.: Coordinating multiple concurrent negotiations. In: The Third International Joint Conference on Autonomous Agents and Multi Agent Systems, New York, USA, pp. 1062–1069 (2004)
13. Rahwan, I., Kowalczyk, R., Pham, H.H.: Intelligent agents for automated one-to-many e-commerce negotiation. In: Twenty-Fifth Australian Computer Science Conference, Melbourne, Australia, pp. 197–204 (2002)
14. Raiffa, H.: The Art and Science of Negotiation. Harvard Univiversity Press, Cambridge (1982)
15. Zhang, Z., Cheng, T., Wu, B., Yang, S.: Price-Based Negotiation for Task Assignment in a Distributed Network Manufacturing Mode Environment. Int. J. Adv. Manuf. Technol. 21, 145–156 (2003)

A Logic for Reasoning about Actions and Explicit Observations

Gavin Rens[1,2], Ivan Varzinczak[1], Thomas Meyer[1,2], and Alexander Ferrein[3]

[1] KRR, CSIR Meraka, Pretoria, South Africa
[2] School of Computer Science, University of KwaZulu-Natal, South Africa
{grens,ivarzinczak,tmeyer}@meraka.org.za
[3] Robotics and Agents Research Lab, University of Cape Town, South Africa
alexander.ferrein@uct.ac.za

Abstract. We propose a formalism for reasoning about actions based on multi-modal logic which allows for expressing observations as first-class objects. We introduce a new modal operator, namely $[o \mid \alpha]$, which allows us to capture the notion of perceiving an observation given that an action has taken place. Formulae of the type $[o \mid \alpha]\varphi$ mean 'after perceiving observation o, given α was performed, necessarily φ'. In this paper, we focus on the challenges concerning sensing with explicit observations, and acting with nondeterministic effects. We present the syntax and semantics, and a correct and decidable tableau calculus for the logic.

1 Introduction and Motivation

Imagine a robot that is in need of an oil refill. There is an open can of oil on the floor within reach of its gripper. If there is nothing else in the robot's gripper, it can grab the can (or miss it, or knock it over) and it can drink the oil by lifting the can to its 'mouth' and pouring the contents in (or miss its mouth and spill). The robot may also want to confirm whether there is anything left in the oil-can by weighing its contents with its arm. And once holding the can, the robot may wish to replace it on the floor.

The domain is (partially) formalized as follows. The robot has the set of (intended) actions $\mathfrak{A} = \{grab, drink, weigh, replace\}$ with expected meanings. The robot can perceive observations only from the set $\Omega = \{obsNil,\ obsHeavy,\ obsMedium,\ obsLight\}$. Intuitively, when the robot performs a *weigh* action, it will perceive either *obsHeavy*, *obsMedium* or *obsLight*; for other actions, it will 'perceive' *obsNil*, no perception. The robot experiences its world (domain) via three Boolean features: $\mathfrak{P} = \{full, drank, holding\}$ meaning respectively that the the oil-can is full, that the robot has drunk the oil and that it is currently holding something in its gripper. This formalization seems more intuitive than lumping all observations in with propositions, for instance, by making $\mathfrak{P} = \{full, drank, holding, obsnil, heavy, medium, light\}$.

It is the norm in dynamic logics (and some other agent oriented logics) to deal with observations as elements of knowledge, as propositions; and perception is normally coded as action, that is, observations-as-propositions evaluate to 'true'

J. Li (Ed.): AI 2010, LNAI 6464, pp. 395–404, 2010.

or 'false' depending on some action(s). However, the approach of interpreting observations as mere propositions may be counterintuitive to some people, because knowledge may be seen as something different from events (observations) that *generate* or *modify* knowledge.

Remark 1. If an intelligent agent is regarded as a *system*, then there are inputs to the system that affect it, and outputs from the system that affect the environment. The inputs are *observations* and the outputs are *actions*. If one assumes that the system state is represented by a knowledge base of *propositions*, then from the systems view, it is clear that observations and propositions are different in nature.

Therefore, the ability to distinguish between observations and propositions allows for a more precise specification of a given domain, as we shall see in the sequel. It turns out that the notion of observations as explicit syntactic and semantic objects of a logic is not completely new. For example, Van Benthem, Gerbrandy and Kooi [15] do so (For more details, the reader is invited to see Section 5 on related work.)

Although there are several formalisms in the literature on reasoning about and specifying agents and their actions, we found them lacking when it comes to treating observations as objects on a par with actions, while retaining important computational properties. Existing first-order based approaches are in general undecidable or have too complicated semantics. For these reasons, we prefer to anchor our framework on a version of dynamic logic and strive for an extension of it by allowing for observations as explicit entities.

The rest of this paper is organized as follows. We give the syntax and semantics of our logic in Section 2. In Section 3, we show how to correctly specify agent domains with our logic. Our tableau method, with correctness and decidability results, is given in Section 4. Section 5 covers related work and Section 6 concludes the paper.

2 A Logic for Actions and Observations

The logic we present here allows for expressing observations explicitly, distinct from propositions. It is called the Logic for Actions and Observations (LAO). LAO is a non-standard modal logic with quantification and equality over the actions and observations. It will be able to accommodate formal descriptions of nondeterminism in the actions and of uncertainty in the observations. Given a formalization $\mathcal{K}$ of our scenario, the robot may have the following queries:

- Is it possible that after grabbing the oil-can, I will not be holding it? That is, does $\langle grab\rangle\neg holding$ follow from $\mathcal{K}$?
- If I weigh the oil-can and perceive that it is heavy, is it necessary that I have drunk the oil? That is, does $[obsHeavy \mid weigh]\, drank$ follow from $\mathcal{K}$?

LAO is based on $\mathcal{LAP}$ (the Logic for Actions and Plans [1]), but with one major difference: the addition of *observations*. That is, LAO refers to a set of

observations that are explicitly identified by a knowledge engineer or agent-system designer (cf. Remark 1). A minor, yet important difference is the addition of action and observation variables, quantification and equality.

2.1 Syntax

We work in a propositional language. It contains three sorts: (1) a finite set of *fluents* (alias *propositional atoms*) $\mathfrak{P} = \{p_1, \ldots, p_n\}$, (2) a finite set of names of atomic *actions* $\mathfrak{A} = \{\alpha_1, \ldots, \alpha_n\}$ and a countable set of *action variables* $V_{\mathfrak{A}} = \{v_1^{\alpha}, v_2^{\alpha}, \ldots\}$, and (3) a finite set of names of atomic *observations* $\Omega = \{\varsigma_1, \ldots, \varsigma_n\}$ and a countable set of *observation variables* $V_{\Omega} = \{v_1^{\varsigma}, v_2^{\varsigma}, \ldots\}$. We shall refer to elements of $\mathfrak{A} \cup \Omega$ as *constants* and elements of $V_{\mathfrak{A}} \cup V_{\Omega}$ as *variables*. A *literal* ℓ is a fluent or its negation.

We are going to work in a multi-modal setting, in which we have a modal operator $[\alpha]$, one for each element in $\mathfrak{A}$; and a modal operator $[\varsigma|\alpha]$, one for each pair (α, ς) in $\mathfrak{A} \times \Omega$.

Definition 1. *Let* $\alpha, \alpha' \in (\mathfrak{A} \cup V_{\mathfrak{A}})$, $\varsigma, \varsigma' \in (\Omega \cup V_{\Omega})$, $v \in (V_{\mathfrak{A}} \cup V_{\Omega})$ *and* $p \in \mathfrak{P}$. *The language of LAO, denoted* $\mathcal{L}_{LAO}$, *is the least set of those* φ *that contain no free variables:*

$$\varphi ::= p \mid \top \mid \neg\varphi \mid \varphi \wedge \varphi \mid \alpha = \alpha' \mid \varsigma = \varsigma' \mid [\alpha]\varphi \mid [\varsigma \mid \alpha]\varphi \mid (\forall v)\varphi.$$

For example, $[v^{\varsigma}|\alpha]$ is not in $\mathcal{L}_{LAO}$, but $(\forall v^{\varsigma})[v^{\varsigma}|\alpha]$ is.

As usual, we treat $\bot, \vee, \rightarrow, \leftrightarrow, \neq$ and $\exists$ as abbreviations. The sentence $[\varsigma \mid \alpha]\varphi$ is read 'φ must hold after ς is observed, given α is executed'. For instance, $[\mathit{obsLight} \mid \mathit{weigh}]\neg\mathit{full}$ means 'After perceiving that the oil-can is light, given a weighing action, the can is necessarily not full'. $[\alpha]\varphi$ is read 'φ must hold (after any/every observation) given α is executed'. For instance, $[\mathit{replace}]\neg\mathit{holding}$ means 'After replacing the oil-can, it is definitely not being held (regardless of observations)'. $\langle\alpha\rangle\varphi$ and $\langle\varsigma \mid \alpha\rangle\varphi$ abbreviate $\neg[\alpha]\neg\varphi$ and $\neg[\varsigma \mid \alpha]\neg\varphi$ respectively. One conventional reading for $\langle\alpha\rangle\varphi$ is 'It is possible that φ holds after α is performed'. The reading of $\langle\varsigma \mid \alpha\rangle\varphi$ is 'It is possible that φ holds after ς is perceived, given α is performed'.

We say that a formula is *static* if it mentions no actions.

We write $\varphi|_c^v$ to mean the formula φ with all variables v appearing in it replaced by constant c of the right sort (action or observation).

2.2 Semantics

Our semantics follows that of multi-modal logic K [11]. However, structures (alias, possible worlds models) are non-standard. Intuitively, when talking about some world w, we mean a set of features (*fluents*) that the agent understands and that describes a state of affairs in the world or that describes a possible, alternative world. Let $w : \mathfrak{P} \longrightarrow \{0, 1\}$ be a total function that assigns a truth value to each fluent. Let S be the set of all possible functions w. We call S the *conceivable worlds*.

Definition 2. *A LAO structure is a tuple* $\mathcal{S} = \langle W, R, O, N, Q \rangle$ *such that*

1. $W \subseteq S$ *is a non-empty (finite) set of* possible worlds*;*
2. R *is a mapping that provides an accessibility relation* $R_\alpha : W \longrightarrow W$ *for each action* $\alpha \in \mathfrak{A}$*;*
3. O *is a non-empty finite set of observations;*
4. $N : \Omega \longrightarrow O$ *is a total bijection that associates to each name in* Ω*, a unique observation in* O*;*
5. Q *is a mapping that provides a perceivability relation* $Q_\alpha : O \longrightarrow W$ *for each action* $\alpha \in \mathfrak{A}$*;*
6. *For all* w, w', α*, if* $(w, w') \in R_\alpha$ *then there is an* o *s.t.* $(o, w') \in Q_\alpha$*, for* $w, w' \in W$*,* $\alpha \in \mathfrak{A}$ *and* $o \in O$*.*

R_α defines which worlds w^+ are accessible via action α performed in world w^- and Q_α defines which observations o are perceivable in worlds w^+ accessible via action α. For $\varsigma \in \Omega$, $N(\varsigma) = o \in O$. Because N is a total bijection, it follows that $|O| = |\Omega|$.

Item 6 of Definition 2 implies that actions and observations always appear in pairs, even if implicitly. For example, if action *open-eyes* is performed, several signals are possible, depending on the situation, like *wall-3-meters-ahead* or *overcast-sky*. If the agent performs an action like *step-once-forward*, there is only one observation possible, viz. *null*, the 'dummy' observation. Unlike the eye, the leg (or wheel) is not a sensory organ. When our agent activates a device (the agent acts) and the device receives no input signal, it interprets this state of affairs as the *null* observation (the *null* observation will be denoted by the special named constant, *obsNil*). For every action an agent performs, the agent perceives exactly one observation. This is the approach of POMDPs that we rely on in the present work [10].

Definition 3 (Truth Conditions). *Let* $\mathcal{S}$ *be a LAO structure, with* $\alpha, \alpha' \in \mathfrak{A}$*,* $v^\alpha \in V_\mathfrak{A}$*,* $\varsigma, \varsigma' \in \Omega$*,* $v^\varsigma \in V_\Omega$ *and* $p \in \mathfrak{P}$*. And let* φ *be any sentence in* $\mathcal{L}_{LAO}$*. We say* φ *is* satisfied *at world* w *in structure* $\mathcal{S}$ *(written* $\mathcal{S}, w \models \varphi$*):*

1. $\mathcal{S}, w \models p$ iff $w(p) = 1$*;*
2. $\mathcal{S}, w \models \top$ for any $w \in W$*;*
3. $\mathcal{S}, w \models \neg\varphi$ iff $\mathcal{S}, w \not\models \varphi$*;*
4. $\mathcal{S}, w \models \varphi \wedge \varphi'$ iff $\mathcal{S}, w \models \varphi$ and $\mathcal{S}, w \models \varphi'$*;*
5. $\mathcal{S}, w \models \alpha = \alpha'$ iff $\alpha, \alpha' \in \mathfrak{A}$ are the same element*;*
6. $\mathcal{S}, w \models \varsigma = \varsigma'$ iff $\varsigma, \varsigma' \in \Omega$ are the same element*;*
7. $\mathcal{S}, w \models [\alpha]\varphi$ iff for all w' and o, if $(w, w') \in R_\alpha$ and $(o, w') \in Q_\alpha$ then $\mathcal{S}, w' \models \varphi$*;*
8. $\mathcal{S}, w \models [\varsigma \mid \alpha]\varphi$ iff for all w', if $(w, w') \in R_\alpha$ and $(N(\varsigma), w') \in Q_\alpha$ then $\mathcal{S}, w' \models \varphi$*;*
9. $\mathcal{S}, w \models (\forall v^\alpha)\varphi$ iff $\mathcal{S}, w \models \varphi|^{v^\alpha}_\alpha$ for all $\alpha \in \mathfrak{A}$*;*
10. $\mathcal{S}, w \models (\forall v^\varsigma)\varphi$ iff $\mathcal{S}, w \models \varphi|^{v^\varsigma}_\varsigma$ for all $\varsigma \in \Omega$*.*

A formula φ is true (valid) in a LAO structure (denoted $\mathcal{S} \models \varphi$) if $\mathcal{S}, w \models \varphi$ for every $w \in W$. φ is LAO-valid (denoted $\models_{LAO} \varphi$) if φ is true in every structure $\mathcal{S}$. φ is *satisfiable* if $\mathcal{S}, w \models \varphi$ for some $\mathcal{S}$ and $w \in W$. We define *global logical entailment* (denoted $\psi \models_G \varphi$) as follows: for all $\mathcal{S}$, if $\mathcal{S} \models \psi$, then $\mathcal{S} \models \varphi$.

The motivation behind the definition of $\mathcal{S}, w \models [o \mid \alpha]\varphi$ is as follows. Just as φ needs not hold in worlds w' if $(w, w') \notin R_\alpha$, worlds w' are not considered if $(o, w') \notin Q_\alpha$. In other words, whether or not a world w' is reachable (via R_α), if the agent perceived o and the agent knows that o is not perceivable in w', then the agent knows it is not in w'. Then what is true or false in w' has no influence on the verity of $\mathcal{S}, w \models [o \mid \alpha]\varphi$. But in every world w' reachable from w *and* in which o is perceivable, φ must be true. While actions can add worlds that an agent believes possible, thus increasing uncertainty, observations eliminate reachable worlds from consideration, thus increasing certainty.

Proposition 1. $\models_{LAO} (\forall v^\alpha)\langle v^\alpha\rangle\varphi \rightarrow (\exists v^\varsigma)\langle v^\varsigma \mid v^\alpha\rangle\varphi$.

This means that for any structure $\mathcal{S}$ and world w, for any action α, if world w' can be reached from w via α, then there exists an observation perceivable in w'. Proposition 1 follows from item 6 of Definition 2.

Due to the nature of the 'observation naming' function N, in the rest of this paper, in our intuitive explanations, we let o mean o or ς (such that $N(\varsigma) = o$) depending on the context, and similarly we let ς mean ς or o.

3 Specifying Domains in LAO

In this section we address how to formally specify the domain in which an agent or robot is expected to live, in the language of LAO. Here, ϕ—with or without subscripts—denotes some (pre)condition expressed as a static sentence.

Firstly, axioms are required for action outcomes that say when an action is executable and for observations that say when (in which worlds) an observation is perceivable. A fundamental assumption in the reasoning about actions and change (RAC) community is that there must be one *executability axiom* for each action type. Executability axioms are similar to the precondition axioms in Reiter's situation calculus [12]. In multi-modal logic, one writes $\langle\alpha\rangle\top \leftrightarrow \phi$ to mean that it is possible to perform α if and only if the precondition ϕ holds. For instance, $\langle grab\rangle\top \leftrightarrow \neg holding$, $\langle drink\rangle\top \leftrightarrow holding \wedge full$, $\langle weigh\rangle\top \leftrightarrow holding$ and $\langle replace\rangle\top \leftrightarrow holding$ define in what worlds it is possible to execute each of the four available actions. There must be an executability axiom for each action in $\mathfrak{A}$.

We follow a systematic approach to specifying domain axioms that is based on the approach of Demolombe, Herzig and Varzinczak [3], which is in turn related to Reiter's approach with functions to systematize the specification of successor-state axioms in the situation calculus [12]. Here, there is only enough space to show the result of the systematic approach.

Effect axioms are required to capture the effects of actions. For the robot scenario, there exists an effect axiom $\neg holding \rightarrow (\langle grab\rangle holding \wedge \langle grab\rangle \neg holding)$

for the *grab* action. A translation for this axiom is, 'There exists an observation such that, if I am not holding the oil-can, it is possible that either I will be holding it or will still not be holding it, after grabbing it.' *grab* is a nondeterministic action with respect to *holding*. And the systematic approach produces the following effect axiom for *drink*: $full \wedge holding \rightarrow [drink]\neg full$.

Frame axioms and *condition closure axioms* state when actions do not have effects—these can easily be expressed in LAO [13].

In the same vein as executability axioms, we need *perceivability axioms*. However, to explain their specification, we define *ontic* (physical) actions and *sensory* actions. Ontic actions have intentional ontic effects, that is, effects on the environment that were the main intention of the agent. Sensory actions result in perception, and might only have (unintentional) side-effects.

Ontic actions (α_{ont}) each have a perceivability axiom of the form

$$(\forall o)\langle o \mid \alpha_{ont}\rangle\top \leftrightarrow o = obsNil.$$

For ontic actions, the null observation is perceived if and only if the action is executed. *grab*, *drink* and *replace* are ontic actions:

$$(\forall o)\langle o \mid grab\rangle\top \leftrightarrow o = obsNil;$$
$$(\forall o)\langle o \mid drink\rangle\top \leftrightarrow o = obsNil;$$
$$(\forall o)\langle o \mid replace\rangle\top \leftrightarrow o = obsNil.$$

For any instantiation of an observation o' other than *obsNil*, according to the semantics, $[o' \mid \alpha_{ont}]\bot$ is a logical consequence of these axioms.

Sensory actions typically have multiple observations and associated conditions for perceiving them. Sensory actions (α_{sen}) thus each have a set of perceivability axioms of the form

$$\langle o_1 \mid \alpha_{sen}\rangle\phi_1, \langle o_2 \mid \alpha_{sen}\rangle\phi_2, \ldots, \langle o_n \mid \alpha_{sen}\rangle\phi_n,$$

for stating when the associated observations are *possible*, where $\{o_1, o_2, \ldots, o_n\} = Dom(Q_{\alpha_{sen}})$[1] and the ϕ_i are the conditions. The following axioms state when observations associated with α_{sen} are *impossible*:

$$\neg\langle o_1 \mid \alpha_{sen}\rangle\neg\phi_1, \neg\langle o_2 \mid \alpha_{sen}\rangle\neg\phi_2, \ldots, \neg\langle o_n \mid \alpha_{sen}\rangle\neg\phi_n.$$

Note that the ϕ_i conditions need not characterize pair-wise disjoint sets of worlds, because more than one observation is allowed in the same worlds, given some action (see, e.g., *obsLight* and *obsMedium* below).

Lastly, to state that the observations not associated with action α_{sen} are always impossible given α_{sen} was executed, we need an axiom of the form

$$(\forall o)(o \neq o_1 \wedge o \neq o_2 \wedge \ldots \wedge o \neq o_n) \rightarrow \neg\langle o \mid \alpha_{sen}\rangle\top,$$

for each action.

[1] $Dom(Q_\alpha)$ is the set of all first elements in the pairs that make up Q_α.

Our only sensory action is *weigh* and its behavior with respect to perception can be captured by the following sentences.

$$\langle obsLight \mid weigh\rangle(\neg full \vee drank) \wedge [obsLight \mid weigh](\neg full \vee drank);$$
$$\langle obsHeavy \mid weigh\rangle(full \vee \neg drank) \wedge [obsHeavy \mid weigh](full \vee \neg drank);$$
$$\langle obsMedium \mid weigh\rangle\top \wedge [obsMedium \mid weigh]\top;$$
$$(\forall o)(o \neq obsHeavy \wedge o \neq obsLight \wedge o \neq obsMedium) \rightarrow \neg\langle o \mid weigh\rangle\top.$$

For instance, *obsLight* is perceivable given *weigh* was performed, if and only if either the oil-can is empty or the oil has been drunk.

All this can be done for deterministic and nondeterministic effects of actions [13]. All the axioms discussed in this section concern the dynamics of an environment. They are collectively the action laws and are here represented by LAW. All axioms in LAW are *global*, that is, true in every possible world. The state of affairs that an agent is in initially, can be characterized by a static, non-global sentence KB (knowledge base). The main task in LAO is to determine whether an arbitrary sentence φ is implied by KB, given LAW, that is, whether $LAW \models_G KB \rightarrow \varphi$. The next section shows how this can be done.

4 Tableaux for LAO

The tableau calculus we propose is adapted from Castilho, Gasquet and Herzig [1]. It is based on labeled formulae. It is a procedure to determine whether $\mathcal{K} \models_G \Psi$, where $\mathcal{K}$ is any set of global axioms in $\mathcal{L}_{LAO}$ and Ψ is any sentence in $\mathcal{L}_{LAO}$.

The tableau calculus for LAO, with all its rules, remarks and observations, is referred to as $\mathcal{C}_{LAO}$. The set of formulae to be checked (i.e., the initial set of formulae to which $\mathcal{C}_{LAO}$ must be applied) is called the *trunk*. A *labeled formula* is a pair (n, φ), where φ is a formula and n is an integer from the set of whole numbers, called the *label* of φ. A *skeleton* Σ is a ternary relation $\Sigma \subseteq (\Omega \cup \mathbb{N}) \times \mathfrak{A} \times \mathbb{N}$. Elements $(\cdot, a, n')$ of the relation are denoted $\cdot \xrightarrow{a} n'$. A *tree* $\mathcal{T}^i$ is a pair $\langle \Gamma^i, \Sigma^i\rangle$, where Γ^i is a set of labeled formulae and Σ^i is a skeleton. The initial tree is $\mathcal{T}^0 = \langle\{(0, \neg\Psi)\}, \emptyset\rangle$. Each $\mathcal{T}^{i+1}$ may be obtained from $\mathcal{T}^i$ by applying certain tableau rules to $\mathcal{T}^i$. Other rules add elements to Γ^i or Σ^i, producing a new state of the tree, but not necessarily a new tree.

Let a particular state of a tree be called a *node*. The application of a rule to a node k results in a new node k'. k' may be a node of the same tree as k, or k' may be the first node of a new tree. A *tableau* for the trunk is a set of trees $\mathcal{T}^0, \ldots, \mathcal{T}^n$ and their states, resulting from the application of *tableau rules* to the trunk and subsequent nodes. The tableau rules for LAO are:

- rule $\perp$: If Γ contains (n, φ) and $(n, \neg\varphi)$, then add $(n, \perp)$ to it.
- rule $\neg$: If Γ contains $(n, \neg\neg\varphi)$, then add (n, φ) to it.
- rule $\wedge$: If Γ contains $(n, \varphi \wedge \varphi')$, then add (n, φ) and (n, φ') to it.
- rule $\vee$: If Γ contains $(n, \neg(\varphi \wedge \varphi'))$, then add $(n, \neg\varphi)$ to it, and create $\mathcal{T}^i = \langle\Gamma \cup \{(n, \neg\varphi')\}, \Sigma\rangle$, where i is a new integer.

- rule $=$: If Γ contains $(n, c = c')$ and in fact, constants c and c' do not refer to the same constant, then add $(n, \bot)$ to it.
- rule $\forall$: If Γ contains $(n, (\forall v)\varphi)$ then add $(n, \varphi|_c^v)$ to it only if the constant c (of the right sort) appears in a formula in Γ.
- rule $\exists$: If Γ contains $(n, \neg(\forall v)\varphi)$ then add $(n, \neg\varphi|_{c_1}^v \vee \ldots \vee \neg\varphi|_{c_j}^v)$ to Γ, for each constant in $\{c_1, \ldots c_j\}$ (of the right sort) that appears in the vocabulary.
- rule $\langle\alpha\rangle$: If Γ contains $(n, \neg[\alpha]\varphi)$, then add $(n, \neg(\forall o)[o \mid \alpha]\varphi)$ and $(n', \neg\varphi)$ to it, and add $n \xrightarrow{\alpha} n'$ to Σ, where n' is a fresh integer. For each $\beta \in \mathcal{K}$, add (n', β) to Γ.
- rule $[\alpha]$: If Γ contains $(n, [\alpha]\varphi)$ and Σ contains $n \xrightarrow{\alpha} n'$, add (n', φ) to Γ.
- rule $\langle o \mid \alpha\rangle$: If Γ contains $(n, \neg[o \mid \alpha]\varphi)$, then add $(n', \neg\varphi)$ to it, and add $n \xrightarrow{\alpha} n'$ and $o \xrightarrow{\alpha} n'$ to Σ, where n' is a fresh integer. For each $\beta \in \mathcal{K}$, add (n', β) to Γ.
- rule $[o \mid \alpha]$: If Γ contains $(n, [o \mid \alpha]\varphi)$ and Σ contains $n \xrightarrow{\alpha} n'$ and $o \xrightarrow{\alpha} n'$, then add (n', φ) to Γ.

The addition of $(n, \neg(\forall o)[o \mid \alpha]\varphi)$ to Γ in rule $\langle\alpha\rangle$ is due to Proposition 1. To make explicit that the formulae in $\mathcal{K}$ are global, they are all added to each new world (fresh integer) introduced in rules $\langle\alpha\rangle$ and $\langle o \mid \alpha\rangle$.

A tree $\langle\Gamma, \Sigma\rangle$ is *closed* if $(i, \bot) \in \Gamma$ for some i. It is *open* if it is not closed. A tableau is closed if all of its trees $\mathcal{T}^0, \ldots, \mathcal{T}^n$ are closed, else it is open.

Definition 4. *If a tableau for $\neg\Psi$ is closed (under $\mathcal{K}$), we write $\mathcal{K} \vdash_{LAO} \Psi$. If there is a* saturated *open tableau for $\neg\Psi$, we write $\mathcal{K} \nvdash_{LAO} \Psi$. A tableau is saturated if any rule that can be applied has been applied to all open trees.*

Theorem 1. *$\mathcal{C}_{LAO}$ is sound (if $\mathcal{K} \vdash_{LAO} \Psi$ then $\mathcal{K} \models_G \Psi$), complete (if $\mathcal{K} \models_G \Psi$ then $\mathcal{K} \vdash_{LAO} \Psi$) and decidable ($\mathcal{C}_{LAO}$ always terminates). [13]*

Using $\mathcal{C}_{LAO}$, the following can be proven:

- $LAW \models_G (full \wedge \neg drank \wedge \neg holding) \rightarrow \langle grab\rangle \neg holding$;
- $LAW \models_G (full \wedge \neg drank \wedge holding) \rightarrow (\exists o)[o \mid drink]\neg full$; and
- $LAW \models_G (full \wedge \neg drank \wedge holding) \rightarrow (\forall o)\langle o \mid drink\rangle \neg full$.

5 Discussion and Related Work

We believe that calculi based on first-order logic, like the situation calculus [9] and the event calculus [7] are too rich for our needs. We thus sought a simpler logic with the potential of being decidable. $\mathcal{LAP}$, the Logic of Actions and Plans, was found to be a suitable basis for our work. $\mathcal{LAP}$ is a multi-modal logic, close to but simpler than Propositional Dynamic Logic [5]. Castilho, Gasquet and Herzig [1] claim that it is sufficient to express most of the problems investigated in the field, however, it does not deal with sensing. To say that LAO is an extension of $\mathcal{LAP}$ is too strong. For example, their definition of $[\alpha]$ is the standard one for multi-modal logic, whereas the definition of $[\alpha]$ in LAO is not standard, in that in LAO its definition involves the *perceivability* relation Q_α.

A series of articles exists concerning probabilistic dynamic epistemic logic (PDEL) [6,14,15], which add probabilistic notions to dynamic epistemic logic (DEL) [16]. The language of PDEL includes formulae of the form $[A, e]\varphi$, where A is a "probabilistic update model" and e is an "event" from the domain of A. The terms *event* and *observation* often have the same meaning in probability theory. Observations in probability theory do not describe a state, but capture information about natural *occurrences*. The authors [15] allude that their events are closer to observations than to logical propositions. Therefore, as far as observations go, PDEL's $[A, e]\varphi$ corresponds to LAO's $[o \mid \alpha]\varphi$, however, the semantics of PDEL's operator is much richer. Note though, that PDEL is an epistemic logic, not a logic about action.

For our work, we have also found some inspiration from the language $\mathcal{ES}$ of Lakemeyer and Levesque [8], especially because $\mathcal{ES}$ has been extended to $\mathcal{ESP}$ [4] to include notions of probability—in a current line of investigation, we also intend to extend LAO with notions of probability. Although it is a situation-based logic, $\mathcal{ES}$ does not include situation terms. It is a second-order modal dialect with *object* and *action* sorts, and with universal quantification and equality. It has fluent and rigid functions and predicates. Fluent predicates include the special predicate *Poss* for defining preconditions on action executability, and a special predicate symbol for defining whether a sensing action was successful. The formula $[\alpha]\varphi$ in $\mathcal{ES}$ is defined to mean 'after α, φ is true'. The meaning is similar to that of the standard modal logic operator, although, in $\mathcal{ES}$, actions are deterministic. The diamond operator is not defined, but with *Poss* available, it needs not be defined. $\mathcal{ES}$'s $[\alpha]$ is thus also different to LAO's $[\alpha]$.

6 Concluding Remarks

Modal logic based RAC formalisms lack a straight-forward way to deal with sensing. In an attempt to solve the problem, we presented a multi-modal logic which includes reasoning about 'reified' observations on a par with actions. It was shown how to specify an agent domain in the language. We provided a tableau calculus ($\mathcal{C}_{LAO}$) for determining the validity of sentences of the logic, and it was stated that the calculus is sound, complete and decidable.

By adding observations to a simple dynamic logic explicitly, the resulting logic may be slightly more complex, while perhaps simplifying, for the domain expert, dealing with explicit observations (cf. Remark 1). Computational complexity of $\mathcal{C}_{LAO}$, and the influence of 'reifying' observations, must still be established. Since LAO is at least as expressive as multi-modal logic K and entailment here is global, we know that LAO is at least EXPTIME.

One of the main problems in systems for RAC is the *frame problem*. We have formulated a solution for LAO, which involves universal quantification over actions. For the interested reader, our *frame solution* appears in the accompanying technical report [13]. Alternatively, because LAO is essentially based on $\mathcal{LAP}$ one could in the future, adapt the frame solution from Castilho, Herzig and Varzinczak [2] to LAO.

Our next aim is to extend LAO to allow one to express uncertainty in action and perception by providing the machinery to specify *probabilistic* models (descriptions) for action outcomes and for perceivability of observations.

Acknowledgments

Thanks to Arina Britz, Gerhard Lakemeyer and the referees for their suggestions.

References

1. Castilho, M., Gasquet, O., Herzig, A.: Formalizing action and change in modal logic I: The frame problem. Journal of Logic and Computation 9(5), 701–735 (1999)
2. Castilho, M., Herzig, A., Varzinczak, I.: It depends on the context! A decidable logic of actions and plans based on a ternary dependence relation. In: 9th International Workshop on Nonmonotonic Reasoning, NMR 2002 (2002)
3. Demolombe, R., Herzig, A., Varzinczak, I.: Regression in modal logic. Journal of Applied Non-Classical Logics 13(2), 165–185 (2003)
4. Gabaldon, A., Lakemeyer, G.: $\mathcal{ESP}$: A logic of only-knowing, noisy sensing and acting. In: Proc. of 22nd Natl. Conf. on Artificial Intelligence (AAAI 2007), pp. 974–979. AAAI Press, Menlo Park (2007)
5. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press, Cambridge (2000)
6. Kooi, B.: Probabilistic dynamic epistemic logic. Journal of Logic, Language and Information 12(4), 381–408 (2003)
7. Kowalski, R., Sergot, M.: A logic-based calculus of events. New Generation Computing 4, 67–95 (1986)
8. Lakemeyer, G., Levesque, H.: A semantic characterization of a useful fragment of the situation calculus with knowledge. In: Special issue in honor of John McCarthy, Artificial Intelligence, Elsevier, Amsterdam (2010)
9. McCarthy, J., Hayes, P.: Some philosophical problems from the standpoint of artificial intelligence. Machine Intelligence 4, 463–502 (1969)
10. Monahan, G.E.: A survey of partially observable Markov decision processes: Theory, models, and algorithms. Management Science 28(1), 1–16 (1982)
11. Popkorn, S.: First Steps in Modal Logic. Cambridge University Press, Cambridge (1994)
12. Reiter, R.: Knowledge in action: logical foundations for specifying and implementing dynamical systems. MIT Press, Massachusetts (2001)
13. Rens, G., Varzinczak, I., Meyer, T.: Partially observable Markov decision logic, Part I: Introducing explicit observation constants, KRR-10-01. Tech. rep., Knowledge Representation and Reasoning, Meraka Institute, CSIR, Pretoria, South Africa (June 2010), http://krr.meraka.org.za/publications/2010
14. Sack, J.: Extending probabilistic dynamic epistemic logic. Synthese 169, 124–257 (2009)
15. Van Benthem, J., Gerbrandy, J., Kooi, B.: Dynamic update with probabilities. Studia Logica 93(1), 67–96 (2009)
16. Van Ditmarsch, H., Van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Springer, Dordrecht (2007)

Taking Learning Out of Real-Time Heuristic Search for Video-Game Pathfinding

Ramon Lawrence[1] and Vadim Bulitko[2]

[1] University of British Columbia Okanagan
ramon.lawrence@ubc.ca
[2] University of Alberta
bulitko@ualberta.ca

Abstract. Real-time heuristic search algorithms are useful when the amount of time or memory resources are limited or a rapid response time is required. An example of such a problem is pathfinding in video games where numerous units may be simultaneously required to react promptly to player's commands. Classic real-time heuristic search algorithms cannot be deployed due to their obvious state-revisitation ("scrubbing"). Recent algorithms have improved performance by using a database of pre-computed subgoals. However, a common issue is that the pre-computation time can be large, and there is no guarantee that the pre-computed data adequately covers the search space. In this work, we present a new approach that guarantees coverage by abstracting the search space using the same algorithm that performs the real-time search. It reduces the pre-computation time via the use of dynamic programming. The new approach has a fast move time and eliminates learning and "scrubbing". Experimental results on maps of millions of cells show significantly faster execution times compared to previous algorithms.

1 Introduction

As search problems become larger, the amount of memory and time to produce an optimal answer using standard search algorithms such as A* [5] increases substantially. This is an issue in resource-limited domains such as video game pathfinding. In real-time search, the amount of planning time per move is bounded independently of the problem size. This is useful when an agent does not have time to compute the entire plan before making a move. Recent real-time search algorithms such as D LRTA* [3], kNN LRTA* [2], and TBA* [1] satisfy the real-time constraint. D LRTA* and kNN LRTA* both use pre-computed subgoal databases to guide the search. However, as the search space grows, the pre-computation time is prohibitively long in practice.

In this paper, we describe a new real-time search algorithm called HCDPS (Hill Climbing and Dynamic Programming Search) that outperforms previous state-of-the-art algorithms by requiring less pre-computation time, having faster execution times, and eliminating state-revisitation. This contribution is achieved with two ideas. First, instead of using a generic way of partitioning the map (e.g., into cliques [10] or sectors [9]), we partition the map into reachability regions. The reachability is defined with respect to the underlying pathfinding algorithm which guarantees that when traversing within such regions, our agent can never get stuck. This fact allows us to replace a learning algorithm (e.g., LRTA* [7]) with simple greedy hill climbing. Doing so simplifies the

J. Li (Ed.): AI 2010, LNAI 6464, pp. 405–414, 2010.

algorithm, eliminates scrubbing, and allows a minimal online memory footprint per agent which, in turn, enables many more agents to path find simultaneously.

The second idea is an applying dynamic programming to database pre-computation. Once we partition the map into regions of hill-climbing reachability, we use dynamic programming to approximate optimal paths between representatives of any two such regions. This is in contrast to computing optimal paths for all region pairs with A* [3]. In our experiments, the benefits of this approximation are substantial: a two orders of magnitude speed-up in the database pre-computation time. In summary, in the domain of pathfinding on maps of over ten million states, HCDPS takes about five minutes of pre-computation per map, has a path suboptimality of about 10%, a move time of $0.23\mu s$, and overall execution time two orders of magnitude faster than A* and TBA*.

2 Problem Formulation

We define a heuristic search problem as a directed graph containing a finite set of states and weighted edges and two states designated as *start* and *goal*. At every time step, a search agent has a single *current state*, a vertex in the search graph which it can change by taking an action (i.e., traversing an out-edge of the current state). Each edge has a positive cost associated with it. The total cost of edges traversed by an agent from its start state until it arrives at the goal state is called the *solution cost*. We require algorithms to be *complete* (i.e., produce a path from start to goal in a finite amount of time if such a path exists). We adopt the standard assumption of safe explorability of the search space (i.e., there are no reachable vertices with in-edges only).

In principle, all algorithms in this paper are applicable to any such heuristic search problem. However, the presentation and experimental evaluation focus on pathfinding on grid-based video game maps. In such settings, states are vacant square grid cells. Each cell is connected to four cardinally and four diagonally neighboring cells. Out-edges of a vertex are moves available in the cell, and we use the terms *action* and *move* interchangeably. The edge costs are 1 for cardinal moves and 1.4 for diagonal moves.

An agent plans its next action by considering states in a local search space surrounding its current position. A *heuristic function* (or simply *heuristic*) estimates the optimal travel cost between a state and the goal. It is used by the agent to rank available actions and select the most promising one. We consider only *admissible* and *consistent* heuristic functions which do not overestimate the actual remaining cost to the goal and whose difference in values for any two states does not exceed the cost of an optimal path between these states. In grid maps we use the standard *octile distance* as our heuristic. The octile distance uses 1 and 1.4 as the edge costs and is equivalent to the optimal travel cost on a map without walls. An agent can modify or *learn* its heuristic function to improve its action selection with experience.

The defining property of real-time heuristic search is that the amount of planning the agent does per action has an upper bound that does not depend on the total number of states in the problem space. We measure the *move time* as the mean planning per action in terms of CPU time. The second performance measure of our study is *sub-optimality* defined as the ratio of the solution cost found by the agent to the optimal solution cost minus one and times 100%. To illustrate, suboptimality of 0% indicates an optimal path and suboptimality of 50% indicates a path 1.5 times as costly as the optimal path.

3 Related Work

Many search algorithms such as A*, IDA* [6] and PRA* [9] cannot guarantee a constant bound on planning time per action as they produce a complete solution before the first action is taken. As the problem size increases, the planning time and corresponding response time will exceed any set limit. Real-time search algorithms repeatedly interleave *planning* (i.e., selecting the most promising action) and *execution* (i.e., performing the selected action). This allows actions to be taken without solving the entire problem which improves response time at the potential cost of suboptimal solutions. LRTA* was the first algorithm and updates/learns its heuristic function with experience. The learning process may make the agent "scrub" (i.e., repeatedly re-visit) the state space to fill in heuristic local minima or *heuristic depressions* [8]. This degrades solution quality and is a show-stopper for video game pathfinding.

Improved performance is possible by pre-computing path information. In its precomputation phase, D LRTA* abstracts the search problem using the clique abstraction of PRA* [10] and then builds a database of optimal paths between all pairs of ground-level representatives of distinct abstract states. The database does not store the entire path but only the ground-level state where the path enters the next region. Online, the agent repeatedly queries the database to identify its next subgoal and runs LRTA* to it. The issues with D LRTA* are the large amount of memory used and the lengthy precomputation time. Further, D LRTA* repeatedly applies the clique abstraction thereby creating large irregular regions. As a result, membership of every ground state to the regions has to be explicitly stored which takes up as much memory as the search problem. Additionally, the abstract regions can contain local heuristic depressions thereby trapping the underlying LRTA* agent and causing learning and scrubbing.

kNN LRTA* attempts to address D LRTA*'s shortcomings by not using a state abstraction and instead pre-computing a set number of optimal paths between randomly selected pairs of states. On each optimal path, the farthest state that is still reachable from the path beginning via hill climbing is then stored as a subgoal. Online, a kNN LRTA* agent uses its database in an attempt to find a similar pre-computed path and then runs LRTA* to the associated subgoal. While kNN LRTA* is more memory efficient than D LRTA*, its random paths do not guarantee that a suitable pre-computed path will be found for a given problem. In such cases, kNN LRTA* runs LRTA* to the global goal which subjects it to heuristic depressions and the resulting learning and scrubbing. Additionally, pre-computing D LRTA* and kNN LRTA* databases is time-consuming (e.g., over a hundred hours for a single video game map).

TBA* forgoes LRTA* learning and runs a time-sliced version of A*. It does not precompute any subgoals and has to "fill in" heuristic depressions online with its open and closed lists. Thus, it consumes more memory per agent and is slower per move.

Our algorithm combines the best features of the previous algorithms. Like D LRTA* and kNN LRTA*, we run our real-time agent toward a near-by subgoal as opposed to a distant global goal, but we also guarantee that any problem will indeed have a suitable series of subgoals each of which is reachable from the preceding one via simple hill climbing. Like TBA*, we do not store or update heuristic values thereby simplifying the implementation, eliminating any scrubbing and saving memory. Unlike TBA*, we also do not use memory for open and closed lists.

4 Intuition for Our Approach

HCDPS operates in two stages: offline and online. Offline, it analyzes its search space and pre-computes a database of subgoals. The database covers the space such that any pair of start and goal states will have a suitable series of subgoals in the database. This is accomplished by abstracting the space. We partition the space into regions in such a way that any state in the region is mutually reachable via simple hill climbing with a designated state, called the *representative* of the region. Since the abstraction builds regions using hill climbing which is also used in the online phase, we are guaranteed that for any start state a our agent can hill climb to a region representative of some region A. Likewise, for any goal state b, there is a region B that the goal falls into which means that the agent will be able to hill climb from B's representative to b. All we need now is a hill-climbable path between the representative of region A and the representative of region B. Unlike canonical states used to derive better heuristics [11], our region representatives are used as subgoals. Unlike visibility polygons used in robot pathfinding, our regions are applicable to arbitrary search graphs as well as grids.

We could pre-compute such paths by running A* between representatives of any two regions on the map. However, this pre-computation is expensive as it scales quadratically with the number of regions. To speed up the pre-computation, we adopt a different approach. Specifically, for every pair of immediately neighboring regions, we run A* in the ground-level space to compute an optimal path between region representatives. We then use dynamic programming to assemble the optimal paths between immediately neighboring regions into paths between more distant regions until we have an (approximately optimal) path between any two regions. To save memory, the resulting paths are compressed into a sequence of subgoals so that each subgoal is reachable from the preceding one via hill climbing. Each such sequence of subgoals is stored as a record in the subgoal database. We then build an index for the database that maps any state to its region representative in constant time.

Online, for a given pair of start and goal states, we use the index to find their region representatives. The subgoal path between the region representatives is retrieved from the database. The agent first hill climbs from its start state to the region representative. We then feed the record's subgoals to the agent one by one until the end of the record is reached. Finally, the agent hill climbs from the region representative to the goal state.

5 Implementation Details

5.1 Offline Stage

The hill-climbing agent used offline and online is a simple greedy search. In its current state s, such an agent considers immediately neighboring states and selects the state s_{next} that minimizes $f(s_{\text{next}}) = g(s, s_{\text{next}}) + h(s_{\text{next}})$ where $g(s, s_{\text{next}})$ is the cost of traversing an edge between s and s_{next} and h is the heuristic estimate of the travel cost between s_{next} and the agent's goal. Ties in f are broken towards higher g. Remaining ties are broken randomly. The agent then moves from s to s_{next} and the cycle repeats. Hill climbing is terminated when a plateau or a local minimum in h is reached: $\forall s_{\text{next}} [h(s) \leq h(s_{\text{next}})]$. If this happens before the agent reaches its goal, we say that the goal is *not* hill-climbing

Fig. 1. Region partitioning of a grid map

reachable from the agent's position. The agent does not use memory for heuristic values or open and closed lists.

Our partitioning mechanism is as follows. Each region R starts with a seed (representative) state r selected among yet unpartitioned states. Then, for each existing state in the region, we form a queue of candidate states to be added. Each candidate state is an immediate neighbor of some existing state in the region. For each candidate state s we check if s is mutually hill-climbing reachable with r and is closer to r than its currently associated seed state. If so, we add s to R. The distance check allows an already assigned state to change regions if its heuristic (octile) distance is closer to another region's seed. Partitioning stops when every ground state is assigned to a region. As the online part of HCDPS starts by heading for the region representative of its start region, we keep the regions fairly small to reduce suboptimality by imposing a cut-off c such that any state assigned to a region is no more than c steps from the region representative. We place seeds regularly along grid axes. In Figure 1 each region is a different color. Region shapes are based on the map properties and the initial starting seeds.

Given the regions and their representatives, we compute approximately optimal paths between all pairs of distinct region representatives with the Floyd-Warshall algorithm [4,12] which incorporates dynamic programming. Specifically, the paths are stored in a two-dimensional array indexed by region numbers. The array is initialized with actual optimal paths computed using A* from each region's representative to representatives of the immediately neighboring regions, or in general a neighborhood depth up to L regions away. We iteratively update elements of the array until they stabilize.

Note that this problem does not exhibit optimal substructure. Specifically, an optimal path between a representative for the region A and a representative for the region B does not necessarily contain optimal ground-level paths between A and C and between

C and B even if the path passes through the region C. Thus, the computed paths are approximations to optimal paths but the savings in time are considerable.

Then we compress each computed path into a series of subgoals. The compression algorithm we use is an extension of the one used in kNN LRTA*. Specifically, given an actual path p, we initialize a compressed path p' by storing the beginning state of p in it. We then use binary search to find the state $s_i \in p$ such that s_i is not hill-climbing reachable from the end state of p' but the immediately preceding state $s_{i-1} \in p$ is. We add s_{i-1} to p' and repeat the process until we reach the end state of p which we then add to p' as well. Each compressed path is a record in our database.

The offline stage finishes with building an index over the database records to allow record retrieval in constant time. A two dimensional array is used to store a path record between each pair of region representatives. Entry (i, j) stores the database record from region representative i to j. Second, the mapping between each ground-level state and its region representative is compressed using run-length encoding (RLE) into an array sorted by state id (a unique scalar). To guarantee constant access time, we build a hash table which maps every k-th state id to its corresponding record in the RLE array. Probing the hash table involves dividing the ground-level state id G by k to get a hash table entry that maps to the RLE entry for ground-level state $\lfloor \frac{G}{k} \rfloor k$. If this RLE range does not contain G, a linear search is performed to find the correct range. In the worst case, this searches k entries if each entry represents only one ground-level state.

As an example, let the RLE table entries be: $(0, 1)$, $(625, 4)$, $(1200, 3)$, $(1600, 1)$, $(2100, 6)$. The first two entries mean that states with ids from 0 to 624 map to region 1. If $k = 1000$, the hash table has three entries: $(0, 0)$, $(1000, 1)$, $(2000, 2)$. The record $(1000, 1)$ means that id 1000 maps to entry 1 in the RLE table which is $(625, 4)$ (indexing starts at 0). Id 1000 maps to region 4 as it falls in the range $[625, 1200)$. State id 1500 maps to hash table entry $\lfloor \frac{1500}{1000} \rfloor = 1$ which is $(1000, 1)$. This gets us to RLE entry $(625, 4)$. 1500 is not in the range $[625, 1200)$ but we scan forward to find RLE entry $(1200, 3)$. Thus, state 1500 is mapped to region 3 as it is in the range $[1200, 1600)$.

5.2 Online Stage

Given a problem $(s_{\text{start}}, s_{\text{goal}})$, the HCDPS agent searches its database to find the record (r_i,r_j) where s_{start} is hill-climbing reachable to r_i and r_j is hill-climbing reachable to s_{goal}. r_i and r_j are region representatives for s_{start} and s_{goal} respectively and have a pre-computed path between them. The agent hill climbs from s_{start} to r_i, follows the subgoals in the path from r_i to r_j, and then from r_j to s_{goal}.

There are several enhancements to this basic process designed to improve solution optimality at the cost of increasing planning time per move. First, we check if the s_{goal} is hill-climbing reachable from s_{start}. If so, then the database is not used at all. Second, when we use a record, we check if its first subgoal is hill-climbing reachable from s_{start}. If so then we direct the agent to go to the first subgoal instead of the record's start. Third, when the agent reaches the last subgoal, it checks if s_{goal} is reachable from its current position. If so then it heads straight for the goal. Otherwise, it goes to the end of the record and then to the goal. Finally, to keep all such checks real-time, we limit the number of hill-climbing steps to a constant, map-independent cutoff c based on the desired response time and the amount of planning time available per move.

6 Theoretical Analysis

HCDPS has several desirable properties including:

1. Guaranteed hill-climbability within a record. For each record (i.e., a compressed path), its first subgoal is hill-climbing reachable from the path beginning. Each subgoal is hill-climbing reachable from the previous one. The end of the path is hill-climbing reachable from the last subgoal.

2. Guaranteed suitable record. For every state s there is a representative r_i state reachable from s via hill climbing. Every pair of region representatives r_i and r_j are connected by a compressed path in the database. Thus, an HCDPS agent can hill climb from s to r_i and then to r_j. From there it can hill climb to its goal state.

3. Completeness. For any solvable problem (i.e., a start and an end state that are reachable from each other), HCDPS will find a path between the start and the end in a finite amount of time with at most visiting any state twice.

Proof. By Property 1, for any problem $(s_{\text{start}}, s_{\text{goal}})$ there is a suitable database record with the start r_i and the end r_j such that r_i is hill-climbing reachable from s_{start} and s_{goal} is hill-climbing reachable from r_j. By Property 2, r_j is hill-climbable from r_i which means that HCDPS can hill climb from s_{start} to r_i to r_j to s_{goal}. Note that there are no state re-visitation within each hill climb. So the only possible state re-visitations can occur when a state visited on the climb from s_{start} to r_i gets re-visited on the climb from r_i to r_j. Likewise, a state visited on the climb from r_i to r_j can be re-visited on the climb from r_j to s_{goal}. □

4. Offline Space Complexity. Let N_R be the number of regions built by HCDPS offline. Then the number of compressed paths in the database is $O(N_R^2)$. Each path is at most $d_{\max}$ states where $d_{\max}$ is the diameter of the space and hence the worst-case database size is $O(d_{\max} N_R^2)$. Mapping between all states and their regions adds $O(N)$ space where N is the number of states. Thus, the total worst-case space complexity is $O(N + d_{\max} N_R^2)$.

5. Offline Time Complexity. An average region has N/N_R states and takes $O(N\sqrt{N/N_R})$ hill climbing steps to build, as a state can be added to at most N_R regions due to the distance check. Thus the total partitioning time is $O(NN_R\sqrt{N/N_R})$. A* is run for no more than $N_R B^L$ problems when each of the N_R regions has no more than B immediately neighboring regions. L is the depth of the neighborhood considered. Thus, the total A* run time is $O(N_R B^L N \log N)$ in the worst case. Running dynamic programming takes $O(N_R^3)$. Each of the resulting N_R^2 paths requires no more than $O(d_{\max} \log d_{\max})$ to compress. Building the compressed mapping table requires a scan of the map and is $O(N)$. Hence the overall worst-case offline complexity is $O(NN_R\sqrt{N/N_R} + N_R B^L N \log N + N_R^3 + N_R^2 d_{\max} \log d_{\max} + N)$.

6. Online Space Complexity. HCDPS uses $O(b)$ for hill climbing where b is the maximum number of neighbors of any state. However, it needs to load the database $O(d_{\max} N_R^2)$ and the index $O(N)$ resulting in the total space complexity of $O(d_{\max} N_R^2 + N)$. Note that the database is shared among $K \geq 1$ of simultaneously pathfinding agents. Thus, per-agent worst-case space complexity is $O(\frac{1}{K}(N + d_{\max} N_R^2))$.

7. Real-timeness. The worst-case online time complexity is $O(b)$. Using the hash table, database query time is $O(k)$.

7 Results

The HCDPS algorithm was compared against D LRTA*, kNN LRTA* and TBA* for pathfinding on game maps from *Counter-Strike: Source* (Valve Corporation), a popular first-person shooter. The grid dimensions varied between 4096×4604 and 7261×4096 cells giving these maps between 18 and 30 million grid cells, which is a two to three orders of magnitude increase in size over most previous papers. We did not compare against LRTA* due to its inferior performance. We did not compare to weighted A*and other approximate search algorithms as they are not real-time. Algorithms were tested using Java 6 under SUSE Linux 10 on an AMD Opteron 2.1 GHz processor.

We used 1000 randomly generated problems across four maps (one such map is in Figure 1). There were 250 problems on each map, and they had a solution cost of at least 1000. For each problem we computed an optimal solution cost by running A*. The optimal cost was in the range of $[1003.8, 2999.8]$ with a mean of 1882, a median of 1855 and a standard deviation of 550. We measured the A* difficulty defined as the ratio of the number of states expanded by A* to the number of edges in the resulting optimal path. For the 1000 problems, the A* difficulty was in the range of $[1, 200]$ with a mean of 63, a median of 36 and a standard deviation of 64.

HCDPS was run for neighborhood depth $L \in \{1, 2, 3, 4, 5\}$. D LRTA* was run with clique abstraction levels of $\{9, 10, 11, 12\}$. kNN LRTA* was run with database sizes of $\{10000, 40000, 60000, 80000\}$ records. We used a cutoff $c = 250$ steps for hill climbing and $k = 1000$ for RLE indexing. kNN LRTA* used reachability checks on the 10 most similar records. TBA* was run with the time slices of $\{5, 10, 50, 100, 500, 1000\}$ states expanded. Its cost ratio of expanding a node to backtracking was set to 10.

We chose the space of control parameters with three considerations. First, we had to cover enough of the space to clearly determine the relationship between control parameters and algorithm's performance. Second, we attempted to establish the *pareto-optimal frontier* (i.e., determine which algorithms *dominate* others by simultaneously outperforming them along two performance measures such as time per move and sub-optimality). Third, we had to be able to run the algorithms in a practical amount of time (e.g., building a database for D LRTA*(8) is not practical as it takes over 800 hours).

7.1 Database Generation

Two measures for database generation are generation time and database size. Generation time, although offline, is important in practice, especially when done on the client side for player-made maps. Database generation statistics averaged per map are given in Table 1. HCDPS is one to two orders of magnitude faster than kNN LRTA* and D LRTA* (levels 9 and 10) in database generation time (**DBTime**). Additionally, HCDPS databases are two orders of magnitude smaller than those of D LRTA* and smaller than kNN LRTA* with better performance (**DBSize**). TBA* does not compute a database.

We vary how many levels of neighbors (L) are considered at the initialization of the Floyd-Warshall algorithm. More levels of neighbors improves the suboptimality performance at the cost of a longer generation time, specifically the A* time to compute paths between additional region pairs. For any value of L tried, partitioning the map, dynamic programming and path compression take about the same amounts of time (140, 0.5

Table 1. Offline and online results

Algorithm	DBTime (hours)	DBSize (KB)	Online mem (KB)	Total mem (KB)	Move time (μs)	Overall Time (ms)	Subopt. (%)
D LRTA* (12)	0.25	87000	19	87019	3.73	1449.37	15999.2
D LRTA* (11)	1.57	87008	11	87019	3.93	814.66	8497.1
D LRTA* (10)	11.95	87058	8	87066	4.26	662.40	6831.7
D LRTA* (9)	89.88	87453	3	87456	3.94	72.38	819.7
kNN LRTA*(10K)	13.10	256	9	265	7.56	665.00	6851.6
kNN LRTA*(40K)	51.89	1029	5	1034	6.88	93.71	620.6
kNN LRTA*(60K)	77.30	1544	4	1548	6.40	11.10	12.9
kNN LRTA*(80K)	103.09	2058	4	2062	6.55	11.30	12.0
TBA*(5)	0	0	1354	1354	14.31	579.77	1504.5
TBA*(10)	0	0	1354	1354	26.34	532.04	666.5
TBA*(50)	0	0	1354	1354	83.31	488.59	131.1
TBA*(100)	0	0	1354	1354	117.52	487.38	64.7
TBA*(500)	0	0	1354	1354	205.92	458.78	11.4
TBA*(1000)	0	0	1354	1354	229.21	459.81	5.3
HCDPS (1)	0.08	2254	0	2254	0.22	0.74	12.1
HCDPS (2)	0.09	2246	0	2246	0.23	0.78	10.6
HCDPS (3)	0.12	2229	0	2229	0.23	0.72	10.1
HCDPS (4)	0.21	2231	0	2231	0.23	0.74	10.0
HCDPS (5)	0.42	2223	0	2223	0.23	0.73	10.0
A*	0	0	1354	1354	335230	335.23	0

and 130 seconds respectively). However, A* takes 4 seconds for immediate neighbors ($L = 1$) and 1250 seconds for a neighborhood of depth $L = 5$. Of the total database size, approximately 58% is for compressed record storage, 38% is for the abstraction index mapping ground-level states to abstract states, and 4% is for the hash table on the abstraction index to guarantee constant time access. The abstraction mapping size is less than 1% of the number of map states. The hash table has an entry for every $k = 1000$ states resulting in a hash table of fewer than 30000 entries.

7.2 Online Performance

As per Table 1, HCDPS is greatly superior to D LRTA* in terms of suboptimality and better than kNN LRTA* as well. Furthermore, it is more robust than kNN LRTA* because it never fails to find a suitable database record and thus never resorts to the global goal. It is also more robust than D LRTA* because its core agent never gets trapped in a heuristic depression within a region. This advantage can be quantified in terms of *maximum* suboptimality over the 1000 test problems: 277000% for D LRTA*(9), 2600% for kNN LRTA*(60000) but only 49% for HCDPS(3).

Online memory is reported as the maximum size of the open and closed lists plus the storage for updated heuristics. HCDPS uses no such memory. Even when considering the total memory (i.e., adding the database size), HCDPS is substantially better than D LRTA* and approximately the same as kNN LRTA* to achieve similar suboptimality performance. It uses about 50% more memory than TBA* and A*. However, the

memory advantage of TBA* and A* disappears with two or more agents pathfinding simultaneously on the same map and sharing the HCDPS database. This is very common in video game pathfinding with anywhere from half a dozen to a thousand agents pathfinding at once. Furthermore, the HCDPS database is read-only which is advantageous on certain hardware such as flash memory.

Finally, HCDPS has the fastest move (response) time of all algorithms and, in particular, is about 60 to 1000 times faster than TBA* and about 1.5 million times faster than A*, which is not a real-time algorithm and needs to compute an entire path before taking the first move. Even if HCDPS is not used as a real-time algorithm, its overall planning time per problem is still around 450 times faster than A*.

8 Conclusion and Future Work Directions

In this work we have presented HCDPS, the first real-time heuristic search algorithm with neither heuristic learning nor maintenance of open and closed lists. Online, HCDPS is simple to implement and dominates the current state-of-the-art algorithms by being simultaneously faster and better in solution quality. It is free of learning and the resulting state re-visitation — which tends to be a show-stopping problem with all previously published real-time search algorithms. This performance is achieved by computing a specially designed database of subgoals. Database pre-computation with HCDPS is two orders of magnitude faster than kNN LRTA* and D LRTA*. Finally, its read-only database gives it a smaller per-agent memory footprint than A* or TBA* with two or more agents. In summary, we feel that HCDPS is presently the best practical real-time search algorithm for video game pathfinding on static maps. Supporting dynamic search spaces by modifying the database in real-time is an avenue of future research.

References

1. Björnsson, Y., Bulitko, V., Sturtevant, N.: TBA*: Time-bounded A*. In: IJCAI, pp. 431–436 (2009)
2. Bulitko, V., Björnsson, Y.: kNN LRTA*: Simple subgoaling for real-time search. In: AIIDE, pp. 2–7 (2009)
3. Bulitko, V., Luštrek, M., Schaeffer, J., Björnsson, Y., Sigmundarson, S.: Dynamic control in real-time heuristic search. JAIR 32, 419–452 (2008)
4. Floyd, R.W.: Algorithm 97: Shortest path. Communications of the ACM 6(5), 345 (1962)
5. Hart, P., Nilsson, N., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. on Sys. Sci. and Cybernetics 4(2), 100–107 (1968)
6. Korf, R.: Depth-first iterative deepening: An optimal admissible tree search. AI 27(3), 97–109 (1985)
7. Korf, R.: Real-time heuristic search. AIJ 42(2-3), 189–211 (1990)
8. Shimbo, M., Ishida, T.: Controlling the learning process of real-time heuristic search. AI 146(1), 1–41 (2003)
9. Sturtevant, N.: Memory-efficient abstractions for pathfinding. In: AIIDE, pp. 31–36 (2007)
10. Sturtevant, N., Buro, M.: Partial pathfinding using map abstraction and refinement. In: AAAI, pp. 1392–1397 (2005)
11. Sturtevant, N.R., Felner, A., Barrer, M., Schaeffer, J., Burch, N.: Memory-based heuristics for explicit state spaces. In: IJCAI, pp. 609–614 (2009)
12. Warshall, S.: A theorem on boolean matrices. J. ACM 1(9), 11–12 (1962)

Heuristic Planning with SAT: Beyond Uninformed Depth-First Search

Jussi Rintanen

NICTA and the Australian National University
Canberra, Australia

Abstract. Planning-specific heuristics for SAT have recently been shown to produce planners that match best earlier ones that use other search methods, including the until now dominant heuristic state-space search. The heuristics are simple and natural, and enforce pure depth-first search with backward chaining in the standard conflict-directed clause learning (CDCL) framework.

In this work we consider alternatives to pure depth-first search, and show that carefully chosen randomized search order, which is not strictly depth-first, allows to leverage the intrinsic strengths of CDCL better, and will lead to a planner that clearly outperforms existing planners.

1 Introduction

Translation into SAT, the satisfiability problem of the classical propositional logic, has been one of the main approaches to solving AI planning problems. The basic idea, first presented by Kautz and Selman [1], is to consider a bounded-horizon planning problem, to represent the values of state variables at every time point as propositional variables, and to represent the relation between two consecutive states as a propositional formula. This idea is essentially the same as in the simulation of nondeterministic polynomial-time Turing machines in Cook's proof of NP-hardness of SAT [2]. Kautz and Selman's idea, and Cook's even more so, was considered to be only of theoretical interest until 1996 when algorithms for SAT had developed far enough to make planning with SAT practical and even competitive with other search methods [3].

Recently, planning-specific improvements to generic SAT algorithms have been proposed. Specifically, the conflict-directed clause learning (CDCL) algorithm for SAT can be forced to do depth-first backward chaining search by a suitable variable selection scheme [4]. Although the idea is very simple and elegant, surprisingly it also results in dramatic improvements to SAT-based planning, and lifts its efficiency to the same level with the currently best algorithms for classical planning [4].

In this paper we propose an alternative search scheme which does not enforce a strict depth-first search. Algorithms for SAT have a great flexibility in choosing the decision variables, and the kind of fixed schemes as in the predecessor work, do not, for most applications, lead to the best possible performance (and for many applications would lead to a poor performance.) The technical challenge is increasing the flexibility in the decision variable selection in a way that actually improves performance.

Additionally, we propose heuristics to order goals and subgoals. The predecessor work [4] ordered them arbitrarily, determined by their order in the input file.

J. Li (Ed.): AI 2010, LNAI 6464, pp. 415–424, 2010.

Our experiments show that with the new improvements our planner substantially outperforms all well-known classical planning systems, including LAMA, the winner of the last (2008) planning competition.

The structure of the paper is as follows. Section 2 explains the background of the work. In Section 3 we present the earlier variable selection scheme. Section 4 extends it by goal-ordering heuristics and by relaxing the search order. In Section 5 we experimentally evaluate the impact of the techniques. We conclude the paper in Section 6.

2 Preliminaries

The classical planning problem involves finding an action sequence from a given initial state to a goal state. The actions are deterministic, which means that an action and the current state determine the successor state uniquely. A state $s : A \to \{0,1\}$ is a valuation of A, a finite set of *state variables*. In the simplest formalization of planning, actions are pairs (p,e) where p and e are consistent sets of propositional literals over A, respectively called *the precondition* and *the effects*. We define $\mathrm{prec}((p,e)) = p$. Actions of this form are known as STRIPS actions for historical reasons. An action (p,e) is *executable* in a state s if $s \models p$. For a given state s and an action (p,e) executable in s, the unique successor state $s' = \mathit{exec}_{(p,e)}(s)$ is determined by $s' \models e$ and $s'(a) = s(a)$ for all $a \in A$ such that a does not occur in e. This means that the effects are true in the successor state and all state variables not affected by the action retain their values. Given an initial state I, a plan to reach a goal G (a set of literals) is a sequence of actions $o_1, \ldots, o_n$ such that $\mathit{exec}_{o_n}(\mathit{exec}_{o_{n-1}}(\cdots \mathit{exec}_{o_2}(\mathit{exec}_{o_1}(I))\cdots)) \models G$.

The basic idea in applying SAT to planning is, for a given set A of state variables, an initial state I, a set O of actions, goals G and a horizon length T, to construct a formula Φ_T such that $\Phi_T \in$ SAT if and only if there is a plan with horizon $0, \ldots, T$. This formula is expressed in terms of propositional variables $a@0, \ldots, a@T$ for all $a \in A$ and $o@0, \ldots, o@T-1$ for all $o \in O$. For a given $t \geq 0$, the valuation of $a_1@t, \ldots, a_n@t$, where $A = \{a_1, \ldots, a_n\}$, represents the state at time t. The valuation of all variables represents a state sequence so that the difference between two consecutive states corresponds to taking zero or more actions. This can be defined in several different ways [5]. For our purposes it is sufficient that the step-to-step change from state s to s' by a set X of actions satisfies the following three properties: *1)* $s \models p$ for all $(p,e) \in X$, *2)* $s' \models e$ for all $(p,e) \in X$, and *3)* $s' = \mathit{exec}_{o_n}(\mathit{exec}_{o_{n-1}}(\cdots \mathit{exec}_{o_2}(\mathit{exec}_{o_1}(s))\cdots))$ for some ordering $o_1, \ldots, o_n$ of X. These conditions are satisfied by all main encodings of planning as SAT [4]. The only encoding not satisfying these conditions (part 1, specifically) is the relaxed $\exists$-step semantics encoding of Wehrle and Rintanen [6].

Given a translation into propositional logic, planning reduces to finding a horizon length T such that $\Phi_T \in$ SAT, and reading a plan from a satisfying assignment for Φ_T. To find such a T, early works sequentially tested Φ_1, Φ_2, and so on, until a satisfiable formula was found. More efficient algorithms exist [7,8].

3 The Variable Selection Scheme

The conflict-directed clause learning (CDCL) algorithm is the basis of most of the currently best SAT solvers in the zChaff family [9]. Introductory presentations of CDCL

algorithms exist [10,11]. The algorithm repeatedly chooses a decision variable, assigns a truth-value to it, and performs inferences with the unit resolution rule, until a contradiction is obtained (the empty clause is derived, or, equivalently, the current valuation falsifies one of the input clauses or derived clauses.) The sequence of variable assignments that led to the contradiction is analyzed, and a clause preventing the repeated consideration of the same assignment sequence is derived and added to the clause set.

The earlier variable selection scheme for planning [4] performed a depth-first search by a stack-based algorithm, finding one action (decision variable) to be used in the CDCL algorithm as the next variable to which a value is assigned. In this section we present two technically simple extensions that allow more flexible traversal orders and the consideration of more than one candidate decision variable. In Section 4 we will utilize these extensions by proposing subgoal ordering heuristics and a more flexible decision variable selection scheme than the strict depth-first one used earlier.

The main challenge in defining a variable selection scheme is its integration in the overall SAT solving algorithm in a productive way. To achieve this, the variable selection depends not only on the initial state, the goals and the actions, represented by the input clauses, but also the current search state of the CDCL algorithm. The algorithm's execution state is characterized by 1) the current set of learned clauses and 2) the current (partial) valuation reflecting the decisions (variable assignments) and inferences (with unit propagation) made so far. Our variable selection scheme only uses part 2 of the execution state, the current partial valuation v.

The earlier variable selection scheme [4] is based on the following observation: each of the goal literals has to be made *true* by an action, and the precondition literals of each such action have to be made *true* by earlier actions (or, alternatively, these literals have to be *true* in the initial state.)

The first step in selecting a decision variable is finding the earliest time point at which a goal literal can become and remain *true*. This is by going backwards from the end of the horizon to a time point t' in which A) an action making the literal *true* is taken or B) the literal is *false* (and the literal is *true* or *unassigned* thereafter.) The third possibility is that the initial state at time point 0 is reached and the literal is *true* there, and hence nothing needs to be done. In case A we have an action already in the plan, and in case B we choose any action that can change the literal from *false* to *true* between t' and $t'+1$ and use it as a decision variable.[1] In case A we push the literals in the precondition into the stack and find supporting actions for them.

In the earlier work it was shown that finding just one action in a depth-first manner is sufficient for an impressive performance [4]. The new algorithm differs from the earlier algorithm in two respects. First, the depth-first search is not terminated after one action is found, but proceeds further (in Fig. 1 all possible candidate actions will be found.) Second, we replace the stack with a priority queue, which enables the use of a heuristic to impose different traversal orders. These two changes are technically trivial, and the challenge is to utilize them in a way that will actually lead to an improved performance.

The extension of the earlier algorithm [4] for computing a set of actions that support currently unsupported top-level goals or preconditions of actions in the current partial plan is given in Fig. 1. For negative literals $l = \neg a$, $l@t$ means $\neg(a@t)$, and for positive

[1] Such an action must exist because otherwise the literal would have to be *false* also at $t'+1$.

literals $l = a$ it means $a@t$. Similarly, we define the valuation $v(l@t)$ for negative literals $l = \neg a$ by $v(l@t) = 1 - v(a@t)$ whenever $v(a@t)$ is defined. For positive literals $l = a$ of course $v(l@t) = v(a@t)$.

```
 1: procedure support(G, O, T, v)
 2: empty the priority queue;
 3: for all l ∈ G do push l@T into the priority queue;
 4: X := ∅;
 5: while the priority queue is non-empty do
 6:    pop l@t from the priority queue;                    (* Take one (sub)goal. *)
 7:    t' := t − 1;
 8:    found := 0;
 9:    repeat
10:       if v(o@t') = 1 for some o ∈ O with l ∈ eff(o)
11:       then                                            (* The subgoal is already supported. *)
12:          for all l' ∈ prec(o) do push l'@t' into the priority queue;
13:          found := 1;
14:       else if v(l@t') = 0 then                        (* Earliest time it can be made true *)
15:          o := any o ∈ O such that l ∈ eff(o) and v(o@t') ≠ 0;
16:          X := X ∪ {o@t'};
17:          for all l' ∈ prec(o) do push l'@t' into the priority queue;
18:          found := 1;
19:       t' := t' − 1;
20:    until found = 1 or t' < 0;
21: end while
22: return X;
```

Fig. 1. Computation of supports for (sub)goals

The procedure in Fig. 1 is the main component of the variable selection scheme for CDCL given in Fig. 2, in which an action is chosen as the next decision variable for

```
1: S := support(G, O, T, v);
2: if S ≠ ∅ then v(o@t) := 1 for any o@t ∈ S;          (* Found an action. *)
3: else
4:    if there are unassigned a@t for a ∈ A and t ∈ {1, ..., T}
5:    then v(a@t) := v(a@(t − 1)) for any a@t with minimal t
6:    else v(o@t) := 0 for any o ∈ O and t ≥ 0 with o@t unassigned;
```

Fig. 2. Variable selection for planning with the CDCL algorithm

the CDCL algorithm if one is available. If none is available, all goals and subgoals are already supported. The current valuation typically is still not complete, and it is completed by assigning unassigned fact variables the value they have in the predecessor state (line 5) and assigning unassigned action variables the value *false* (line 6). The code in Fig. 2 replaces VSIDS as the variable selection heuristic in the CDCL algorithm.

4 Heuristics for Variable Selection

The variable selection scheme, as described in Section 3, has already led to a planner that is very competitive with the best existing planners for the classical planning problem [4]. However, experience from SAT solvers and from the application of SAT solving to planning specifically [12] suggests that the fixed goal-orderings and the strict backward chaining depth-first search do not – although better than generic SAT-solvers [4] – ultimately represent the most efficient form of search in the CDCL context.

First, we will present a goal-ordering heuristic for controlling the priority queue. If only the first action found is returned, the traversal order in the algorithm in Fig. 1 directly determines the ordering in which variables are assigned in the CDCL algorithm.

Second, the search with strict backward chaining will be relaxed. Backward chaining means selecting an action with an effect x given a goal x, and taking the preconditions of the action as new goals, for which further actions are chosen. The search with backward chaining proceeds step by step toward earlier time points (until some form of backtracking will take place.) In the context of CDCL and other SAT algorithms, the search does not have to be directional in this way, and actions less directly supporting the current (sub)goals could be chosen, arbitrarily many time points earlier. The algorithm in Fig. 1 computes a complete set of candidate actions for supporting all goals and subgoals (as opposed to finding only one as in the predecessor work [4]), but randomly choosing one action from this set is not useful, and we need a more selective way of choosing a decision variable.

Next we will consider these two possible areas of improvement, and in each case propose a modification to the basic variable selection scheme which will be shown to lead to substantial performance improvements in Section 5.

4.1 Goal Ordering

We considered two measures according to which (sub)goals $l@t$ are ordered.

1. the maximal $t' < t$ such that $v(l@t') \neq 1$
2. the maximal $t' < t$ such that $v(l@t') = 0$

Above, $v(l@t') \neq 1$ includes the case that $v(l@t')$ is unassigned. In the first case, l gets a higher priority if it must have been made *true* earlier than other subgoals. The most likely plan involves making l first *true*, followed by making the other subgoals *true*. The second case looks at the time when the subgoals must have been *false* the last time. Empirically the best results were obtained with the first. Intuitively, this measure is a better indicator of the relative ordering of the actions establishing different preconditions of a given action.

A key property of these measures is that for every goal or subgoal $l@t$, the new subgoals $l_1@t-1, \ldots, l_n@t-1$ all have a higher priority than their parent $l@t$. This will still lead to depth-first search, but the ordering of the child nodes will be informed.

4.2 Computation of Several Actions

To achieve a less directional form of plan search with CDCL, we decided to compute some fixed number $N = |S|$ of actions (not only $N = 1$ as in [4]) and randomly choose

one $o@t \in S$. In the algorithm in Fig. 1 this means adding a statement that returns S as soon as $|S| = N$. The initial experiments seemed very promising in solving some of the difficult problems much faster. However, the overall improvement was relatively small, and it very surprisingly peaked at $N = 2$.

What happened is the following. For a given top-level goal $l \in G$, several of the first actions that were chosen supported the goals. However, after everything needed to support l was included, the computation continued from the *next unsupported top-level goal*. So at the final stages of finding support for a top-level goal we would be, in many cases, instead selecting supporting actions for other top-level goals, distracting from finding support for l. With $N = 2$ the distraction is small enough to not outweigh the benefits of considering more than one action.

This analysis led us to a second variant, which proved to be very powerful. In this variant we record the time-stamp t of the first action found. Then we continue finding up to N actions, but *stop and exit* if the time-stamp of a would-be candidate action is $\geq t$. With this variant we obtained a substantial overall improvement with higher N. Later in the experiments we use $N = 10$ because the improvement leveled off at $N = 10$.

4.3 Discussion

The good performance of the fixed and uninformed variable selection [4] is due to its focus on a particular action sequence. Any diversion from a previously tried sequence is a consequence of the clauses learned with CDCL. This maximizes the utility of learned clauses, but also leads to the possibility of getting stuck in a part of the search space void of solutions. A remedy to this problem in current SAT solvers is restarts [9]. However, with deterministic search and without VSIDS-style variable (or action) weighting mechanism restarts make no difference. In SAT algorithms that preceded VSIDS, a small amount of randomization was used to avoid getting stuck [13]. However, too large diversion from the previous action sequences makes it impossible to benefit from the clauses learned with CDCL. Hence the key problem is finding a balance between focus to recently traversed parts of the search space and pursuing other possibilities.

The flexible depth-first style search from Section 4.2 provides an interesting balance between focus and variation. The candidate actions all contribute to one specific way of supporting the top-level goals, but because they often don't exactly correspond to an actual plan (except for at the very last stages of the search), varying the order in which they are considered seems to be an effective way of probing the "mistakes" they contain. An additional benefit seems to be that the non-linear ordering in which the candidate actions are used often leaves holes (missing actions) in the incomplete plan, which are immediately filled by unit propagation. For this reason the number of decisions needed in the CDCL algorithm is sometimes much smaller.

5 Evaluation

Our base line in the evaluation is the backward chaining fixed variable-selection scheme introduced in the predecessor work [4]. This scheme was already shown to outperform the standard VSIDS heuristic, both our own implementation and current best implementations in generic SAT solvers, including Precosat and RSAT.

The test material was 968 problem instances from the international planning competitions from 1998 until 2008. Since our variable selection scheme is defined for the restricted STRIPS language only, we chose all the STRIPS problems except for some from the first competitions, nor did we choose benchmarks from an earlier competition if the same domain had been used in a later competition as well.

We used the most efficient known translation from planning into SAT, for the $\exists$-step semantics by Rintanen et al. [5], and solved the problems with the algorithm B of Rintanen et al. [5] with $B = 0.9$, testing horizon lengths $0, 5, 10, 15, \ldots$ and solving a maximum of 18 SAT problems simultaneously.

All the experiments were run in an Intel Xeon CPU E5405 at 2.00 GHz with a minimum of 4 GB of main memory and using only one CPU core. We ran our planner for all of the problem instances, giving a maximum of 300 seconds for each instance. The runtime includes all standard phases of a planner, starting from parsing the PDDL description of the benchmark and ending in outputting a plan. The different variants of the planner are the baseline fixed variant *base* from the earlier paper [4], *o* with the subgoal ordering from Section 4.1 but with only one action found and returned by the procedure call support(G, O, T, v), *m* with random choice from multiple candidate actions from Section 4.2, and *o*+*m* which combines the previous two. The randomization in *m* and *m*+*o* affects the runtimes, but not much: different complete runs of all 968 instances solved couple of instances more or less, depending on whether for some instances the runtime was slightly below or slightly above the 300 second time limit.

We also tested LAMA [14], the winner of the last (2008) planning competition, and ran it with its default settings, except for limiting its invariant computation to a maximum of 60 seconds according to Helmert's instructions, to adjust for the 300 second time limit we used. Due to a bug in one of its components, LAMA is not able to solve the first instance of OPTICAL-TELEGRAPH and the first 13 instances of PHILOSOPHERS (the rest take longer than 300 seconds.)

The results of the experiment are summarized in Table 1. The first column is the number of (solvable) problem instances in each domain. To get an idea of the differences in the runtime behavior of the different variants of the planner, we plotted a curve showing the number of problem instances solved (y axis) with a given timeout limit (x axis), shown in Fig. 3. Overall, the improvements of the new techniques over the baseline planner and LAMA are substantial, no matter which time out limit is considered.

Other well-performing planners in the planning competitions starting from 2000, including FF and YAHSP from the HSP family of planners which use delete relaxation heuristics [15] and LPG-td [16], are overall very close to LAMA (within 1.5 per cent) in terms of number of solved problem instances. These planners solve respectively 786, 775[2] and 779 problem instances in 300 seconds. As an illustration of the overall performance difference, the number of problem instances FF solves in 30000 seconds equals the number for our planner with a 90 second time limit. This means that FF would have to become more than two orders of magnitude faster on average to match the performance of our planner.

[2] YAHSP does not solve the 30 TPP problems because of a parser bug. Fixing this bug would probably lift YAHSP's number close to 805, making it the second fastest planner after ours.

Table 1. Number of problems solved in 300 seconds for each benchmark domain. Average solution times and numbers of actions for instances solved by all.

domain		VSIDS	base	o	m	o+m	LAMA
1998-GRIPPER	20	20	20	20	20	20	20
1998-MPRIME	20	16	18	18	20	20	20
1998-MYSTERY	19	16	17	17	17	17	19
2000-BLOCKS	102	71	85	86	90	90	51
2000-LOGISTICS	76	76	76	76	76	76	76
2002-DEPOTS	22	21	21	22	22	22	16
2002-DRIVERLOG	20	15	20	20	19	19	20
2002-FREECELL	20	4	5	5	12	12	18
2002-ZENO	20	18	20	20	20	20	20
2004-AIRPORT	50	40	42	41	43	42	37
2004-OPTICAL-TELEG	14	14	14	14	14	14	2
2004-PHILOSOPHERS	29	29	29	29	29	29	BUG
2004-PIPESWORLD-NO	50	15	20	20	33	34	44
2004-PSR-SMALL	50	50	49	49	50	50	50
2004-SATELLITE	36	29	32	32	32	32	30
2006-PIPESWORLD	50	9	10	12	21	24	38
2006-ROVERS	40	40	40	40	39	39	40
2006-STORAGE	30	29	30	30	30	30	18
2006-TPP	30	26	26	28	30	30	30
2006-TRUCKS	30	19	29	29	30	30	8
2008-ELEVATORS	30	13	30	30	30	30	30
2008-OPENSTACKS	30	15	11	11	15	15	30
2008-PARCPRINTER	30	30	30	30	30	30	28
2008-PEGSOLITAIRE	30	25	21	27	23	30	29
2008-SCANALYZER	30	19	16	26	21	27	27
2008-SOKOBAN	30	2	4	4	5	5	18
2008-TRANSPORT	30	10	12	12	20	21	28
2008-WOODWORKING	30	30	30	30	30	30	28
total	968	701	757	778	821	838	775
time average		9.68	6.24	5.99	3.60	3.53	12.23
size average		81.53	60.68	60.40	64.61	64.33	66.64

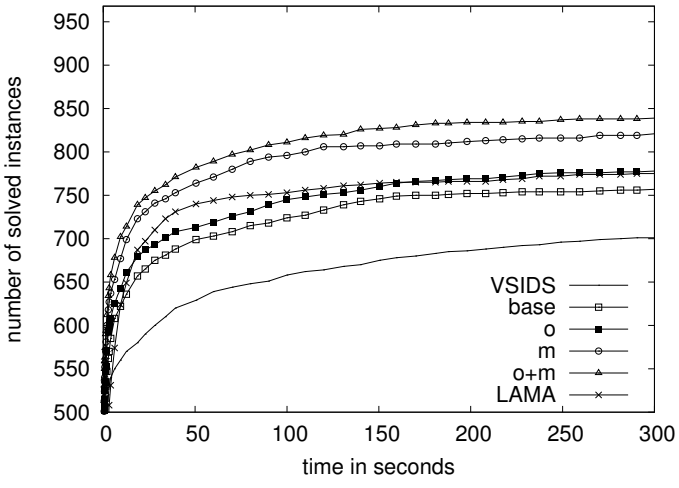


Fig. 3. Number of instances that are solved in a given time

6 Conclusions and Future Work

We have considered a number of goal orderings for a CDCL variable selection scheme for planning, and demonstrated substantial improvements in the performance of SAT solvers in solving standard benchmark problems.

A notable difference between our work and VSIDS [9] is that we are not using weights of decision variables obtained from conflicts as a part of variable selection. Such weights would be able to order the top-level goals and subgoals in the computation of actions, based on their role in conflicts. This, we believe, is the most promising area for future improvement in the implementations of our variable selection scheme.

Acknowledgements

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

1. Kautz, H., Selman, B.: Planning as satisfiability. In: Neumann, B. (ed.) Proceedings of the 10th European Conference on Artificial Intelligence, pp. 359–363. John Wiley & Sons, Chichester (1992)
2. Cook, S.A.: The complexity of theorem proving procedures. In: Proceedings of the Third Annual ACM Symposium on Theory of Computing, pp. 151–158 (1971)
3. Kautz, H., Selman, B.: Pushing the envelope: planning, propositional logic, and stochastic search. In: Proceedings of the 13th National Conference on Artificial Intelligence and the 8th Innovative Applications of Artificial Intelligence Conference, pp. 1194–1201. AAAI Press, Menlo Park (August 1996)

4. Rintanen, J.: Heuristics for planning with SAT. In: Cohen, D. (ed.) CP 2010. LNCS, vol. 6308, pp. 414–428. Springer, Heidelberg (2010)
5. Rintanen, J., Heljanko, K., Niemelä, I.: Planning as satisfiability: parallel plans and algorithms for plan search. Artificial Intelligence 170(12-13), 1031–1080 (2006)
6. Wehrle, M., Rintanen, J.: Planning as satisfiability with relaxed ∃-step plans. In: Orgun, M., Thornton, J. (eds.) AI 2007: Advances in Artificial Intelligence: 20th Australian Joint Conference on Artificial Intelligence, Surfers Paradise, Gold Coast, Australia, December 2-6 (2007)
7. Rintanen, J.: Evaluation strategies for planning as satisfiability. In: López de Mántaras, R., Saitta, L. (eds.) Proceedings of the 16th European Conference on Artificial Intelligence, ECAI 2004, pp. 682–687. IOS Press, Amsterdam (2004)
8. Rintanen, J.: Planning and SAT. In: Biere, A., Heule, M.J.H., van Maaren, H., Walsh, T. (eds.) Handbook of Satisfiability. Frontiers in Artificial Intelligence and Applications, vol. 185, pp. 483–504. IOS Press, Amsterdam (2009)
9. Moskewicz, M.W., Madigan, C.F., Zhao, Y., Zhang, L., Malik, S.: Chaff: Engineering an Efficient SAT Solver. In: Proceedings of the 38th ACM/IEEE Design Automation Conference (DAC 2001), pp. 530–535. ACM Press, New York (2001)
10. Mitchell, D.G.: A SAT solver primer. EATCS Bulletin 85, 112–133 (2005)
11. Beame, P., Kautz, H., Sabharwal, A.: Towards understanding and harnessing the potential of clause learning. Journal of Artificial Intelligence Research 22, 319–351 (2004)
12. Rintanen, J.: A planning algorithm not based on directional search. In: Cohn, A.G., Schubert, L.K., Shapiro, S.C. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Sixth International Conference (KR 1998), pp. 617–624. Morgan Kaufmann Publishers, San Francisco (1998)
13. Gomes, C.P., Selman, B., Kautz, H.: Boosting combinatorial search through randomization. In: Proceedings of the 14th National Conference on Artificial Intelligence (AAAI 1997) and 9th Innovative Applications of Artificial Intelligence Conference (IAAI 1997), pp. 431–437. AAAI Press, Menlo Park (1998)
14. Richter, S., Helmert, M., Westphal, M.: Landmarks revisited. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI 2008), pp. 975–982. AAAI Press, Menlo Park (2008)
15. Bonet, B., Geffner, H.: Planning as heuristic search. Artificial Intelligence 129(1-2), 5–33 (2001)
16. Gerevini, A., Serina, I.: Planning as propositional CSP: from Walksat to local search techniques for action graphs. Constraints Journal 8, 389–413 (2003)

Performance of Infeasibility Empowered Memetic Algorithm (IEMA) on Engineering Design Problems

Hemant K. Singh, Tapabrata Ray, and Warren Smith

School of Engineering and Information Technology,
University of New South Wales, Australian Defence Force Academy, Canberra, ACT
{h.singh,t.ray,w.smith}@adfa.edu.au
http://www.unsw.adfa.edu.au

Abstract. Engineering design optimization problems often involve a number of constraints. These constraints may result from factors such as practicality, safety and functionality of the design and/or limit on time and resources. In addition, for many design problems, each function evaluation may be a result of an expensive computational procedure (such as CFD, FEA etc.), which imposes a limitation on the number of function evaluations that can be carried out to find a near optimal solution. Consequently, there is a significant interest in the optimization community to develop efficient algorithms to deal with constraint optimization problems. In this paper, a new memetic algorithm is presented, which incorporates two mechanisms to expedite the convergence towards the optimum. First is the use of marginally infeasible solutions to intensify the search near constraint boundary, where optimum solution(s) are most likely to be found. Second is performing local search from promising solutions in order to inject good quality solutions in the population early during the search. The performance of the presented algorithm is demonstrated on a set of engineering design problems, using a low computation budget (1000 function evaluations).

Keywords: constraint handling, engineering design, expensive problems.

1 Introduction

In the recent years, population based heuristic algorithms have gained popularity as generic optimizers. This is because they do not require any conditions on continuity or differentiability of objective functions, and hence are suitable for optimization of a wide range of problems. In addition, they can capture the whole Pareto optimal front for multi-objective problems in a single run as opposed to most single point methods.

Most engineering design optimization problems contain a number of constraints. These constraints usually impose limits on space, time, availability of resources, cost, safety and viability of design, aesthetics, ergonomics, and many more. In addition, many of the engineering design problems are *computationally expensive*, which means that evaluating each design (function evaluation) can take a long time.

Artificial intelligence and heuristic optimization techniques are being increasingly used these days to solve a variety of real life optimization problems. However, the usefulness of these applications depend on how efficiently these heuristic methods are

J. Li (Ed.): AI 2010, LNAI 6464, pp. 425–434, 2010.

able to deal with the constraints, especially when there is a limited budget on function evaluations owing to computational complexity. Consequently, constraint handling has attracted a lot of attention from the evolutionary optimization community.

Some of the earlier proposals for constraint handling include widely used penalty function approach, where the objective value is degraded by imposing a penalty on the solutions that violate any of the constraints. A number of variants of penalty functions have been proposed, which include static penalty function models [16], dynamic penalty function models [15], annealing penalty function models [18], adaptive penalty models [11] and death penalty function models [13]. There has also been a number of other proposals which include special representation schemes [3,20], repair strategies [32], separate ranking of objective and constraint functions [9]. Detailed review of various constraint handling techniques used in conjunction with evolutionary algorithms can be found in in [1,19].

Since the final aim of optimization is to achieve the *feasible* optimal solution, a preference for a feasible solution over an infeasible solution is built into the ranking in most of the evolutionary algorithms. Such a preference tries to drive the population towards the feasible search space before improving the objective function(s). However, many a times, the search space may consist of disconnected feasible regions and such a preference may result in localization of solutions in a sub-optimal region, which is not desirable for convergence.

In addition, for most constrained problems, the solution to the optimization problem is likely to lie on the constraint boundary. Therefore, an infeasible solution near the constraint boundary may be more suitable for guiding the search than a feasible solution away from it. Some of the approaches that have exploited information from the infeasible solutions to expedite the search include use of constraints as additional objectives [31,26], explicit parent matching schemes [12], preferential treatment of best infeasible solutions [17] etc.

Recently, Singh *et al.* [30] proposed an infeasibility driven evolutionary algorithm (IDEA), which explicitly maintains marginally infeasible solutions during the search. By maintaining these solutions (in addition to good feasible solutions), the search is intensified near the constraint boundary, where the optimum solution is likely to occur. In addition, the algorithm also provides some marginally infeasible solutions near the optimum solution as an output, which could be used for trade-off studies. The benefit of using infeasibility driven approach over conventional feasible-first ranking procedure was demonstrated on a number of benchmark problems in [30,27].

In the presented work, the algorithm proposed in [30,27] has been further refined in order to find near optimal solutions in relatively fewer function evaluations. In the literature, often global search methods (such as evolutionary algorithms) are used in conjunction with local search methods (such as gradient search) to search for optimum solution efficiently. This *hybrid* approach is referred to as memetic algorithm [21]. For a review on memetic algorithms, the readers are referred to [22]. The algorithm presented in this paper is a memetic algorithm, which uses IDEA as a global search method. Within each generation, a local search is initiated from a promising solution in the population. The primary purpose of the local search is to inject good quality solutions to the population early during the search.

It is worthwhile mentioning here that the use of surrogate models for approximation of objective(s) and constraint(s) can be advantageous for solving computationally expensive problems [14], since the search can be guided using approximations in lieu of the actual function evaluations. The focus of the present work is to improve upon the existing IDEA algorithm by embedding local search in it. The algorithm presented here does not involve surrogate assistance. However, the integration of surrogate modeling with this algorithm will be considered in the future work for further improvements.

Rest of the paper is organized as follows. Since the proposed algorithm utilizes concepts from Infeasibility Driven Evolutionary Algorithm (IDEA), a background on IDEA is given in Section 2. The proposed Infeasibility Empowered Memetic Algorithm (IEMA) is then described in Section 3. The performance of the proposed IEMA on a set of engineering design problems is then reported in Section 4. Finally, a summary of the findings of the paper is presented in Section 5.

2 Infeasibility Driven Evolutionary Algorithm (IDEA)

Infeasibility Driven Evolutionary Algorithm (IDEA) was proposed by Singh *et al.* [30]. It differs from the conventional EAs significantly in the terms of ranking and selection of the solutions. While most EAs rank feasible solutions above infeasible solutions, IDEA ranks solutions based on the original objectives along with additional objective representing constraint violation measure. IDEA explicitly maintains a few infeasible solutions during the search. In addition, "good" infeasible solutions are ranked higher than the feasible solutions, and thereby the search proceeds through both feasible and infeasible regions, resulting in greater rate of convergence to optimal solutions.

The benefits obtained in convergence using explicit preservation of infeasible solutions motivated the development of IDEA [30,27], where the original problem is reformulated as an unconstrained problem with "violation measure" of the solutions as an additional objective. Violation measure is a quantity that is calculated based on the constraint violations of the solutions in he population. The studies reported in [30,27] indicate that IDEA has better rate of convergence compared to a conventional EA for a number of constrained single and multi-objective optimization problems.

A generalized single-objective constrained optimization problem can be formulated as shown in (1)

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) \\ \text{Subject to} \quad & g_i(\mathbf{x}) \geq 0, \quad i = 1,\ldots,m \\ & h_j(\mathbf{x}) = 0, \quad j = 1,\ldots,p \end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1,\ldots,x_n)$ is the design variable vector bounded by lower and upper bounds $\mathbf{x} \in \mathbf{S} \subset \Re^n$. Here, $g(\mathbf{x})$ represents an inequality constraint, whereas $h(\mathbf{x})$ represents an equality constraint. It is a usual practice to convert the equality constraints to inequality constraints using a small tolerance (*i.e.* $h(\mathbf{x}) = 0$ is converted to $|h(\mathbf{x})| \leq \varepsilon$). Hence, the discussion presented here is with regards to presence of inequality constraints only.

To effectively search the design space (including the feasible and the infeasible regions), the original single objective constrained optimization problem is reformulated as bi-objective unconstrained optimization problem as shown in (2).

$$\begin{aligned} \text{Minimize} \quad & f_1'(\mathbf{x}) = f_1(\mathbf{x}) \\ & f_2'(\mathbf{x}) = \text{violation measure} \end{aligned} \tag{2}$$

The additional objective represents a measure of constraint violation, which is referred to as "violation measure". It is based on the amount of relative constraint violations among the population members. Each solution in the population is assigned m ranks, corresponding to m constraints. The ranks are calculated as follows. To get the ranks corresponding to i^{th} constraint, all the solutions are sorted based on the constraint violation value of i^{th} constraint. Solutions that do not violate the constraint are assigned rank 0. The solution with the least constraint violation value gets rank 1, and the rest of the solutions are assigned increasing ranks in the ascending order of their constraint violation values. The process is repeated for all the constraints and as a result each solution in the population gets assigned m ranks. The violation measure of a solution is the sum of these m ranks of the solution corresponding to m constraints.

The main steps of IDEA are outlined in Algorithm 1. IDEA uses simulated binary crossover (SBX) and polynomial mutation operators to generate offspring from a pair of parents selected using binary tournament as in NSGA-II [8]. Individual solutions in the population are evaluated using the original problem definition (1) and the infeasible solutions are identified. The solutions in the parent and offspring population are divided into a feasible set (S_f) and an infeasible set (S_{inf}). The solutions in the feasible set and the infeasible set are ranked separately using the non-dominated sorting and crowding distance sorting [8] based on 2 objectives as per the modified problem definition (2). The solutions for the next generation are selected from both the sets to maintain infeasible solutions in the population. In addition, some of the infeasible solutions are ranked higher than the feasible solutions to provide a selection pressure to create *better* infeasible solutions resulting in an active search through the infeasible search space.

Algorithm 1. Infeasibility Driven Evolutionary Algorithm (IDEA)

Require: N {Population Size}
Require: $N_G > 1$ {Number of Generations}
Require: $0 < \alpha < 1$ {Proportion of infeasible solutions}
1: $N_{inf} = \alpha * N$
2: $N_f = N - N_{inf}$
3: pop_1 = Initialize()
4: Evaluate(pop_1)
5: **for** $i = 2$ to N_G **do**
6: $childpop_{i-1}$ = Evolve(pop_{i-1})
7: Evaluate($childpop_{i-1}$)
8: $(S_f, S_{inf}) = \text{Split}(pop_{i-1} + childpop_{i-1})$
9: Rank(S_f)
10: Rank(S_{inf})
11: $pop_i = S_{inf}(1 : N_{inf}) + S_f(1 : N_f)$
12: **end for**

A user-defined parameter α is used to maintain a set of infeasible solutions as a fraction of the size of the population. The numbers N_f and N_{inf} denote the number of

feasible and infeasible solutions as determined by parameter α. If the infeasible set S_{inf} has more than N_{inf} solutions, then first N_{inf} solutions are selected based on their rank, else all the solutions from S_{inf} are selected. The rest of the solutions are selected from the feasible set S_f, provided there are at least N_f number of feasible solutions. If S_f has fewer solutions, all the feasible solutions are selected and the rest are filled with infeasible solutions from S_{inf}. The solutions are ranked from 1 to N in the order they are selected. Hence, the infeasible solutions selected first are ranked higher than the feasible solutions selected later.

3 Infeasibility Empowered Memetic Algorithm (IEMA)

The proposed algorithm is constructed using IDEA as the baseline algorithm. For single objective problems, a local search can be a very efficient tool for optimization. However, its performance is largely dependent on the starting solution, rendering it unreliable for global optimization. The proposed algorithm tries to exploit the advantages of both these approaches, i.e. a) intensifying the search near the constraint boundary by preserving marginally infeasible solutions and b) effectiveness of local search to expedite the convergence in potentially optimal regions of the search space. As mentioned before, an approach that combines global and local search is termed as a memetic algorithm. Hence, the proposed algorithm as is referred to as Infeasibility Empowered Memetic Algorithm (IEMA).

The proposed IEMA is outlined in algorithm 2. In IEMA, during each generation, apart from the evolution of the solutions using IDEA, a local search is performed from a solution in the population for a prescribed number of function evaluations (set to $20\times n_{var}$ in the presented studies, where n_{var} is the number of design variables). Sequential Quadratic Programming (SQP) [24] has been used in the presented studies for the local search. The starting solution for the local search is determined from the solutions in the population in the following way:

1. If the local search in the previous generation was able to improve the best solution, then the new best solution is used as the starting solution for the local search.
2. If the local search was unable to improve the best solution in the previous generation, it is evident that the existing best solution (in the previous generation) is either not a good starting solution for the local search, or close enough to optimum (either local or global), such that further improvements are difficult. Therefore, in such a case, a random solution is selected from the *high ranked* infeasible solutions *and* the feasible solutions in the population, in an attempt to improve the objective value further. High ranked infeasible solutions consist of the the $N_{inf} = \alpha * N$ solutions (refer to algorithm 2)

After performing the local search the worst solution in the population is replaced by the best solution found from the local search. The ranking of solutions is done in the same way as done in IDEA. The injection of good quality solutions found using the local search guides the population towards potentially optimal regions of the search space. The evolved solutions in turn act as good starting solutions for the local search in subsequent generations. In this way, both IDEA and local search work together to identify the optimum solution.

Algorithm 2. Infeasibility Empowered Memetic Algorithm (IEMA)

Require: N {Population Size}
Require: $N_G > 1$ {Number of Generations}
Require: $0 < \alpha < 1$ {Proportion of infeasible solutions}
1: $N_{inf} = \alpha * N$
2: $N_f = N - N_{inf}$
3: pop_1 = Initialize()
4: Evaluate(pop_1)
5: **for** $i = 2$ to N_G **do**
6: $childpop_{i-1}$ = Evolve(pop_{i-1})
7: Evaluate($childpop_{i-1}$)
8: $(S_f, S_{inf}) = \text{Split}(pop_{i-1} + childpop_{i-1})$
9: Rank(S_f)
10: Rank(S_{inf})
11: $pop_i = S_{inf}(1 : N_{inf}) + S_f(1 : N_f)$
12: $\mathbf{x} \leftarrow$ Choose starting solution in pop_i
13: $\mathbf{x}_{\mathbf{best}} \leftarrow$ Local_search ($\mathbf{x}$) {$\mathbf{x}_{\mathbf{best}}$ is the best solution found using local search from $\mathbf{x}$}
14: Replace worst solution in pop_i with $\mathbf{x}_{\mathbf{best}}$
15: Rank(pop_i) {Rank the solutions again in pop_i}
16: **end for**

4 Numerical Experiments

The performance of the proposed IEMA algorithm is reported on four benchmark engineering design problems, *viz.* Belleville spring design [29], welded beam design [6], car side impact [28] and bulk carrier design [23] (single objective formulation as studied in [30]). These problems have been used in the literature by various researchers in order to test the performance of the constraint handling techniques. The results of IEMA are compared with those obtained from two other algorithms:

1. Non-dominated sorting Genetic Algorithm (NSGA-II) [8], which is one of the most widely used Evolutionary Algorithms for optimization presently.
2. Infeasibility Driven Evolutionary Algorithm (IDEA) [30, 27], which is the precursor to IEMA.

The aim of comparing these three algorithms is to highlight the benefit obtained over NSGA-II by incorporating preservation of good infeasible solutions (IDEA) and then by the further incorporation of local search (IEMA). To this end, percentage improvements in the objective values using these two algorithms over NSGA-II are reported in Table 2. In addition, some of the best results reported earlier in the literature for these problems have also been included for further comparisons.

Very limited number of function evaluations (only 1000) have been allowed for the studies presented here, inline with the paradigm that function evaluations can be often very expensive for engineering design problems.

4.1 Experimental Setup

The crossover and mutation parameters are kept same for all the three algorithms (NSGA-II, IDEA and IEMA), and are listed in Table 1. Thirty independent runs are performed on each problem using each algorithm.

Table 1. Parameters used for the experiments

Parameter	Value
Population Size	40
Maximum function evaluations	1000
Crossover Probability	0.9
Crossover index	10
Mutation Probability	0.1
Mutation index	20
Infeasibility Ratio (α)	0.2

4.2 Results

The results using the three algorithms are summarized in Table 2. It is seen that IEMA is able to achieve better objective values than both IDEA and NSGA-II for all problems. The results of IDEA are better than NSGA-II for all problems except bulk carrier design, for which it is marginally worse (less than 1 %) than NSGA-II.

The percentage improvement attained in using IEMA and IDEA over NSGA-II varies for different problems, but it can be seen that as high as 20.43 % improvement over the best result was obtained using IEMA (for the case of Belleville spring design). Furthermore, the improvements in the median values indicates that IEMA is able to achieve the good objective values very consistently. Again, for the case of Belleville spring design, 47.25 % improvement was seen in the median value using IEMA as compared to that obtained using NSGA-II. The improvements in the the other problems are comparatively less in magnitude, but still significant and consistent.

Another impressive feature of the performance from IEMA for the studied problems is its ability to obtain good objective values in much fewer evaluations as compared to the those reported earlier in literature. In the summary of results shown in Table 2, the function evaluations used in some of the previous studies are also listed, in addition to the best values reported. Except the recent studies by Isaacs [14] which also use 1000 evaluations for comparison, the number of function evaluations used in most other studies are much higher than those used here. Even so, the objectives values reported here are better (or very close) than the best reported previously[1]. Also worth mentioning here is that the best results reported for Belleville spring design and Welded beam design in [14] use surrogate assisted algorithms, but superior results have been obtained in the presented studies without the use of surrogates. This also highlights a further scope of improvement over current studies, *i.e.* inclusion of surrogate assisted techniques in IEMA. The best design vectors found using IEMA are listed in Table 3.

Although the results obtained using the proposed IEMA are very promising, it is not without limitations. The most prominent limitation of IEMA (at least in the current implementation) is its inability to handle discrete variables (during the local search). Therefore, experiments have been reported only on problems with continuous variables. However, it could be resolved with use of more specialized operators. Secondly, the performance is also likely to deteriorate if the number of variables is very high, because the calculation of gradients itself will become computationally expensive in that case.

[1] Please note that slight variations in the results might also result from different precision of the variables or machines used for conducting previously reported experiments.

Table 2. Results for engineering design problems. The numbers in the brackets indicate percent improvement in the objective values compared to those obtained using NSGA-II. (Note: For Belleville spring design, the thickness of the spring has been considered as a discrete variable in [7], but as a continuous variable in others, including the presented studies.)

		IEMA	IDEA	NSGA-II	Other best reported (reference) (evals)
Belleville spring [29]	Best	**1.97967** (20.43 %)	2.20176 (11.50%)	2.48789	2.121964(Coello [2]) (24K)
	Median	**1.97967** (47.25 %)	3.38646 (9.76%)	3.75291	2.29 (Isaacs [14]) (1K)
	Worst	6.32532	7.2774	6.5233	2.16256(Deb,Goyal [7]) (10K)
	std.	0.788324	1.28952	0.978053	1.978715 (Siddall [29]) (infeas.)
	Feasible runs	30	21	21	
Welded beam [6]	Best	**2.38096** (5.48%)	2.45567 (2.52%)	2.51916	2.3854347 (Ray,Liew [25])(33K)
	Median	**2.38096** (33.55%)	2.81411 (21.46%)	3.58301	2.44 (Isaacs [14])(1K)
	Worst	4.69066	4.45493	5.11578	2.38119 (Deb [5])(40K)
	std.	0.560984	0.545042	0.71464	2.38119 (Deb [4]) (320K)
	Feasible runs	30	30	30	
Car side impact [28]	Best	**23.5857** (1.26 %)	23.6988 (0.79%)	23.8872	23.585651 (Saxena,Deb [28])
	Median	**23.5857** (2.90%)	24.0132 (1.14%)	24.2895	23.59 (Gu *et al.* [10])
	Worst	23.5857	25.2929	26.679	
	std.	5.25507e-08	0.368223	0.671836	
	Feasible runs	30	30	30	
Bulk carrier design [23]	Best	**8.60617** (3.47%)	8.93236 (-0.18%)	8.91589	8.6083 (Singh *et al.* [30])(25K)
	Median	**8.72483** (9.43%)	9.70404 (-0.73%)	9.63375	
	Worst	13.1018	11.791	11.827	
	std.	1.56552	0.848578	0.889552	
	Feasible runs	24	30	30	

Table 3. The best design vectors found using the proposed IEMA

Problem	**x**	f
Belleville spring	(12.01, 10.0305, 0.204143, 0.2)	1.97967
Welded beam	(0.244369, 6.21752, 8.29147 , 0.244369)	2.38096
Car side impact	(0.5, 1.22573 , 0.5, 1.20711, 0.875, 0.884189, 0.4,0.345 , 0.192 , 0 , 0)	23.5857
Bulk carrier design	(280.908, 18.4985 , 25.4265, 0.75, 46.8181, 14)	8.60617

5 Summary and Future Work

In this paper, an Infeasibility Empowered Memetic Algorithm (IEMA) is presented. IEMA combines the advantages of IDEA, which focuses the search near the constraint boundaries, with local search, an efficient tool for solving single objective continuous problems. In the proposed algorithm, ranking is done similar to IDEA, and in each generation, the solutions are enhanced by doing a local search from a good quality solution in the population. The performance of IEMA is studied on a set of constrained engineering design problems for a low number of function evaluations. The proposed IEMA offers significant improvements over NSGA-II in objective values for the problems studied, and also compare favorably to the other reported results in the literature. Further improvements in IEMA by incorporating the use of surrogate assisted techniques and enhancements for handling discrete variables are currently underway.

Acknowledgment

The presented work was supported by grants from DSARC, UNSW@ADFA, Australia. The authors would also like to thank Dr. Amitay Isaacs for his support in implementation of the presented algorithm.

References

1. Coello Coello, C.A.: Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Computer Methods in Applied Mechanics and Engineering 191(11-12), 1245–1287 (2002)
2. Coello Coello, C.A.: Treating constraints as objectives for single-objective evolutionary optimization. Engineering Optimization 32(3), 275–308 (2000)
3. Davis, L. (ed.): Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York (1991)
4. Deb, K.: Optimal design of a welded beam via genetic algorithms. AIAA Journal 29(8), 2013–2015 (1991)
5. Deb, K.: An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering 186, 311–338 (2000)
6. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley and Sons Pvt. Ltd., Chichester (2001)
7. Deb, K., Goyal, M.: A combined genetic adaptive search (geneas) for engineering design. Computer Science and Informatics 26, 30–45 (1996)
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)
9. Deb, K.: An Efficient Constraint Handling Method for Genetic Algorithms. Computer Methods in Applied Mechanics and Engineering 186(2), 311–338 (2000)
10. Gu, L., Yang, R., Tho, C., Makowski, M., Faruque, O., Li, Y.: Optimisation and robustness for crashworthiness of side impact. International Journal of Vehicle Design 26(4), 348–360 (2001)
11. Hadj-Alouane, A.B., Bean, J.C.: A Genetic Algorithm for the Multiple-Choice Integer Program. Operations Research 45, 92–101 (1997)
12. Hinterding, R., Michalewicz, Z.: Your brains and my beauty: parent matching for constrained optimisation. In: Proceedings of 1998 IEEE Conference on Evolutionary Computaion, pp. 810–815 (May 1998)
13. Hoffmeister, F., Sprave, J.: Problem-independent handling of constraints by use of metric penalty functions. In: Fogel, L.J., Angeline, P.J., Bäck, T. (eds.) Proceedings of the Fifth Annual Conference on Evolutionary Programming (EP 1996), pp. 289–294. The MIT Press, San Diego (February 1996)
14. Isaacs, A.: Development of optimization methods to solve computationally expensive problems. Ph.D. thesis, University of New South Wales, Australian Defence Force Academy (UNSW@ADFA), Canberra, Australia (2009)
15. Joines, J., Houck, C.: On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs. In: Fogel, D. (ed.) Proceedings of the First IEEE Conference on Evolutionary Computation, Orlando, Florida, pp. 579–584 (1994)
16. Kuri-Morales, A., Quezada, C.V.: A Universal Eclectic Genetic Algorithm for Constrained Optimization. In: Proceedings 6th European Congress on Intelligent Techniques & Soft Computing, EUFIT 1998, Verlag Mainz, Aachen, Germany, pp. 518–522 (September 1998)

17. Mezura-Montes, E., Coello Coello, C.: A simple multimembered evolution strategy to solve constrained optimization problems. IEEE Transactions on Evolutionary Computation 9(1), 1–17 (2005)
18. Michalewicz, Z.: Genetic Algorithms, Numerical Optimization, and Constraints. In: Eshelman, L.J. (ed.) Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA 1995), pp. 151–158. University of Pittsburgh, Morgan Kaufmann Publishers, San Mateo, California (July 1995)
19. 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. The MIT Press, Cambridge (1995)
20. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Heidelberg (1996)
21. Moscato, P.: On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Tech. Rep. C3P report 826, Caltech Concurrent Computation Program, Caltech, California, USA (1989)
22. Ong, Y.S., Lim, M., Chen, X.: Memetic computation;past, present; future [research frontier]. IEEE Computational Intelligence Magazine 5(2), 24–31 (2010)
23. Parsons, M., Scott, R.: Formulation of multicriterion design optimization problems for solution with scalar numerical optimization methods. Journal of Ship Research 48(1), 61–76 (2004)
24. Powell, M.: A fast algorithm for nonlinearly constrained optimization calculations. In: Watson, G. (ed.) Numerical Analysis, pp. 144–157. Springer, Heidelberg (1978)
25. Ray, T., Liew, K.: Society and civilization: An optimization algorithm based on the simulation of social behavior. IEEE Transactions on Evolutionary Computation 7(4), 386–396 (2003)
26. Ray, T., Tai, K., Seow, K.: Multiobjective design optimization by an evolutionary algorithm. Engineering Optimization 33(4), 399–424 (2001)
27. Ray, T., Singh, H.K., Isaacs, A., Smith, W.: Infeasibility driven evolutionary algorithm for constrained optimization. In: Mezura-Montes, E. (ed.) Constraint Handling in Evolutionary Optimization. Studies in Computational Intelligence, pp. 145–165. Springer, Heidelberg (2009)
28. Saxena, D.K., Deb, K.: Trading on infeasibility by exploiting constraint's criticality through multi-objectivization: A system design perspective. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC 2007), September 25-28, pp. 919–926 (2007)
29. Siddall, J.N.: Optimal engineering design - principles and applications. Marcel Dekker, Inc., New York (1982)
30. Singh, H.K., Isaacs, A., Ray, T., Smith, W.: Infeasibility Driven Evolutionary Algorithm (IDEA) for Engineering Design Optimization. In: Wobcke, W., Zhang, M. (eds.) AI 2008. LNCS (LNAI), vol. 5360, pp. 104–115. Springer, Heidelberg (2008)
31. Vieira, D.A.G., Adriano, R.L.S., Vasconcelos, J.A., Krahenbuhl, L.: Treating constraints as objectives in multiobjective optimization problems using niched pareto genetic algorithm. IEEE Transactions on Magnetics 40(2) (March 2004)
32. Xiao, J., Michalewicz, Z., Trojanowski, K.: Adaptive Evolutionary Planner/Navigator for Mobile Robots. IEEE Transactions on Evolutionary Computation 1(1), 18–28 (1997)

Short-Text Similarity Measurement Using Word Sense Disambiguation and Synonym Expansion

Khaled Abdalgader and Andrew Skabar

Department of Computer Science and Computer Engineering
La Trobe University, Bundoora, Australia
kabdalgader@students.latrobe.edu.au, a.skabar@latrobe.edu.au

Abstract. Measuring the similarity between text fragments at the sentence level is made difficult by the fact that two sentences that are semantically related may not contain any words in common. This means that standard IR measures of text similarity, which are based on word co-occurrence and designed to operate at the document level, are not appropriate. While various sentence similarity measures have been recently proposed, these measures do not fully utilise the semantic information available from lexical resources such as WordNet. In this paper we propose a new sentence similarity measure which uses word sense disambiguation and synonym expansion to provide a richer semantic context to measure sentence similarity. Evaluation of the measure on three benchmark datasets shows that as a stand-alone sentence similarity measure, the method achieves better results than other methods recently reported in the literature.

1 Introduction

Measuring the similarity between small-sized text fragments (e.g., sentences) is a fundamental function in applications such as text mining and text summarization, which usually operate at the sentence or sub-sentence level [1][2]; question answering, where it is necessary to calculate the similarity between a question-answer pair [3][4]; and image retrieval, where we are interested in the similarity between a query and an image caption [5]. Although methods for measuring text similarity have been in existence for decades, most approaches are based on word co-occurrence [6][7]. The assumption here is that the more similar two texts are, the more words they have in common. While this assumption is generally valid for large-size text fragments (e.g., documents)—and hence the widespread and successful use of these methods in information retrieval (IR)—the assumption does not hold for small-sized text fragments such as sentences, since two sentences may be semantically similar despite having few, if any, words in common.

One approach to measuring similarity between two sentences is based on representing the sentences in a reduced vector space consisting only of the words contained in the sentences. For example the sentences “Dogs chase cats” and “Felines kill mice” could be represented respectively as the vectors (1, 1, 1, 0, 0, 0) and (0, 0, 0, 1, 1, 1) in a vector space in which dimensions correspond to the ordered terms ‘cat’, ‘chase’, ‘dog’, ‘feline’, ‘kill’ and ‘mice’, and a vector entry of 1 (0) represents the presence (absence) of the corresponding word in the sentence. However this in itself

J. Li (Ed.): AI 2010, LNAI 6464, pp. 435–444, 2010.

does not solve the problem of lack of word co-occurrence, since vector space similarity measures such as cosine similarity will still yield a value of 0.

To solve this problem, rather than assigning a value of 0 for the vector entry corresponding to an absent word, we can estimate a non-zero value which reflects the extent to which the word is related semantically to the collection of words in the sentence. For example, the vector for the first sentence above has a 0 entry corresponding to 'feline'. By comparing the word 'feline' semantically with the words 'dog', 'chase' and 'cat' (by using a dictionary, for example), we would expect to arrive at a non-zero value, since 'feline' would presumably be found to be semantically related to 'cat', as well as to 'dog' (by virtue of cats and dogs being members of the animal kingdom). This will result in a non-zero value for the fourth entry in the first vector, and hence a non-zero value when the cosine similarity of the resulting vectors is calculated. We refer to vectors determined in this way as *semantic vectors.* There are different approaches to measuring the similarity between a word, x, and a collection of n words. One approach is to take the mean semantic similarity between x and each of the n words; another is to use the maximum word-to-word similarity score, which is the approach taken in Li *et al.* (2006) [8].

Sentence similarity measures can also be defined in which sentences are not explicitly represented in a vector space. For example, in Mihalcea *et al.* (2006) [9], each word in a sentence is assigned a score determined as the maximum semantic similarity between the word and the words in the opposing sentence. These scores are then weighted by inverse document frequency (idf) values, summed over both sentences, and finally normalized, resulting in a measure of sentence similarity.

Although the above approaches differ in how the final similarity score is calculated, both are based on the estimation of similarity between a word and a set of words, and the purpose of this paper is to explore how this measurement can be improved through better utilising the semantic information available from lexical resources such as WordNet [10]. The contribution of the paper is two-fold. Firstly, the paper proposes a method by which word sense identification, used in conjunction with synonym expansion, can be used to create an enriched semantic context, enabling a more accurate estimate of semantic similarity. Results of applying the measure to three benchmark datasets shows that as a stand-alone measure, the method achieves better results than other methods recently reported in the literature. The second contribution of the paper is a novel word sense disambiguation (WSD) algorithm that operates by comparing WordNet glosses of the target word with a context vector comprising the remaining words in the sentence.

The remainder of this paper is structured as follows. Section 2 introduces the method of synonym expansion through word sense identification. Section 3 describes the WSD algorithm that we have developed, and the word-to-word semantic similarity measures that we use. Section 4 provides empirical results, and Section 5 concludes the paper.

2 Word Sense Disambiguation and Synonym Expansion

The approach that we present in this section is depicted in Figure 1. For each of the sentences being compared, we first apply a word sense disambiguation step to identify the sense in which words are being used within the sentence. We then apply a synonym expansion step, allowing a richer semantic context from which to estimate

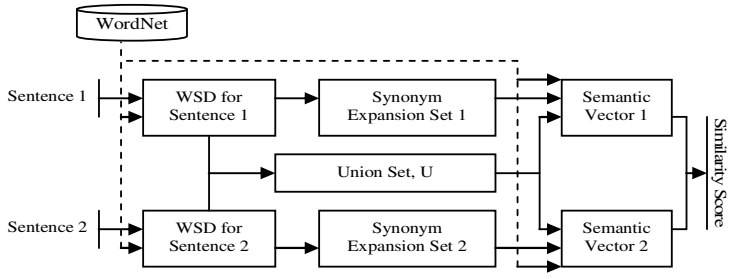


Fig. 1. Sentences Similarity Computation Diagram

semantic vectors. The similarity between semantic vectors can then be calculated using a standard vector space similarity measure such as cosine similarity. We first describe the role of WSD. We then describe the procedure for synonym expansion.

2.1 The Role of Word Sense Disambiguation (WSD)

Sentence similarity as measured using methods such as those described in the Introduction is based on word-to-word similarities. The standard approach used within sentence similarity measures based on WordNet [10] is to simply use the first WordNet sense for each of the two words being compared [8][9]. (Senses in WordNet are ordered from most-frequent to least-frequent). However this can lead to inaccurate similarity measurements. To illustrate, consider the following:

Sentence 1: I deposited a cheque at the bank.
Sentence 2: There is oil sediment on the south bank of the river.

Using the reduced vector space representation we obtain

S1: [('deposited', 0), ('cheque', 0), ('bank', 0)]
S2: [('oil', 0), ('sediment', 0), ('south', 0), ('bank', 0), ('river', 0)]
U: [('river', 0), ('south', 0), ('oil', 0), ('sediment', 0), ('deposited', 0), ('cheque', 0), ('bank', 0)]
V1: [0.066, 0.062, 0.058, 0.055, 1.0, 1.0, 1.0]
V2: [1.0, 1.0, 1.0, 1.0, 0.0, 0.059, 1.0]

where S1 and S2 contain the word-sense pairs for non-stopwords (stopwords are words such as ‘a’, ‘the’, etc, and are removed because they carry little semantic information); U is the reduced vector space, consisting of all word-sense pairs in the union of S1 and S2; and V1 and V2 are the vectors for S1 and S2 in this reduced vector space. The entry corresponding to a word *x* in V1 (V2) is determined as the maximum similarity between *x* and the words in S1 (S2). For example, 0.066 is the maximum similarity between ('river', 0) and the words in S1 having the same part of speech. (Many WordNet word-to-word similarity measures are only defined between words with the same part of speech). Calculating the cosine similarity between V1 and V2 results in a value of 0.33.

The similarity value of 0.33 is likely to be an overestimate. For example, the word ‘bank’ appears in both sentences, but its sense is different in each. Using a WordNet sense of 0 will always result in a maximum similarity between these. Problems might also arise between words which are not common between the two sentences. For example, there is a sense of ‘deposit’ which is closely related to ‘sediment’ (An oil

deposit might be considered a sediment). If we perform the same calculation, but by incorporating WSD we now obtain:

S1: [('deposited', 1), ('cheque', 0), ('bank', 13)]
S2: [('oil', 4), ('sediment', 1), ('south', 3), ('bank', 0), ('river', 0)]
U: [('river', 0), ('bank', 0), ('south', 3), ('deposited', 1), ('sediment', 1), ('cheque', 0), ('oil', 4), ('bank', 13)]
V1: [0.059, 0.051, 0.052, 1.0, 0.044, 1.0, 0.050, 1.0]
V2: [1.0, 1.0, 1.0, 0.050, 1.0, 0.059, 1.0, 0.049]

which results in a cosine similarity value of 0.11. This is lower than that achieved without the use of WSD, and is more in accord with the human judgement that S1 and S2 bare little semantic similarity.

Now consider the following sentences, which most humans would consider to be semantically related:

Sentence 3: The world is in economic crisis.
Sentence 4: The current dismal fiscal situation is global.

Calculating sentence similarity with and without WSD results in similarity values of 0.08 and 0.09 respectively. It is problematic that a value 0.08 has been obtained for a pair of sentences which we consider to be semantically related, yet a higher value of 0.11 was obtained for Sentences 1 and 2, which we consider not to be semantically related. Thus use of WSD on its own appears to be insufficient. In the next section we describe how using synonym expansion can solve this problem.

2.2 Increasing Semantic Context through Synonym Expansion

WordNet [10] represents each unique meaning of a word by a synset consisting of that word together with its synonyms (if any). Synonyms are words with the same meaning. For example, the synset ['fiscal', 'financial'] represents the sense of ('fiscal', 0): "*involving financial matters*". Synsets provides a means of expanding the semantic context. For example, consider Sentences 3 and 4 above. Disambiguating the words in these sentences results in a vector space consisting of the following sense-assigned words:

U: [('fiscal', 0), ('current', 3), ('crisis', 0), ('dismal', 0), ('situation', 0), ('global', 1), ('world', 1), ('economic', 0)]

We can use information from the respective synsets of these words to add context to the original sentences. For example, Sentence 4 above was originally represented as the set

[('current', 3), ('dismal', 0), ('fiscal', 0), ('situation', 0), ('global', 1)]

Using information from the synsets of these words, we can expand this to

[('current', 0), ('blue', 0), ('dark', 0), ('dingy', 0), ('disconsolate', 0), ('gloomy', 0), ('grim', 0), ('sorry', 0), ('drab', 0), ('drear', 0), ('dreary', 0), ('financial', 0), ('state_of_affairs', 0), ('ball-shaped', 0), ('globose', 0), ('globular', 0), ('orbicular', 0), ('spheric', 0), ('spherical', 0)]

It is important to note that all synonyms are added with sense 0. While this might appear counter-intuitive, since this may not be the sense of the synonym in the original synset (i.e., the synset of the word being expanded), it is precisely through including synonyms with sense 0 that we are able to expand the context. There are two inter-related reasons for this. Firstly, adding the correct sense for a synonym would achieve nothing, since the similarity of some word x to this synonym would be the same as its similarity to all other words in the same synset (which includes the identified sense of the original word

used to produce the synset). Secondly, WordNet assigns a sense of 0 to the most frequently used sense of a word. This means that using this sense is most likely (but not guaranteed) to expand the context in a semantic direction of benefit in finding possible semantic similarities between words in the two sentence being compared. We also note that using synonym expansion does not require the dimensionality of the vector space to be increased (i.e., we do not add synonyms to U). The expanded context is utilized when we calculate the semantic vectors. Whereas originally the entries for these vectors was based only on similarities to words in the original sentence, we now consider similarities to the synonyms that have been introduced.

To complete the above example, the expanded description for Sentence 3 is [('domain', 0), ('economic', 0), ('crisis', 0)]. This results in the following semantic vectors:

V3: [0.0, 0.807, 1.0, 0.0, 0.0, 0.068, 0.059, 0.0]
V4: [1.0, 1.0, 0.111, 1.0, 0.0, 0.074, 1.0, 0.0]

These vectors have a cosine similarity of 0.38, which is higher than the value of 0.08 achieved without synonym expansion.

WSD and synonym expansion pull in opposite directions: WSD tends to decrease similarity values; synonym expansion tends to increase them. Thus, even though synonym expansion has increased the similarity value for Sentences 3 and 4, it is likely also to have increased the similarity value for Sentences 1 and 2. While it may appear that WSD and synonym expansion are working at odds, this is not the case. What is crucial to note is that synonym expansion is based on identified word senses. The semantic context is not expanded blindly, but is focused in the vicinity of the semantic context provided by the sense-assigned meanings of the original words. Synonym expansion is not independent from WSD, it requires WSD. We also note that ultimately it is relative—not absolute—similarity values which are important.

In Section 4 we demonstrate empirically that the resulting similarity measure outperforms other recently-proposed measures. We now describe the WSD algorithm we have developed, and the WordNet word-to-word similarity measures which we use.

3 Word Sense Disambiguation

Synonym expansion relies on the correct identification of the WordNet sense in which the word to be expanded is being used. In this section we describe a novel variant of the word sense disambiguation algorithm originally proposed by Lesk (1988) [11].

Lesk's (1988) [11] method determines the sense of a polysemous word by calculating the word overlap between the glosses (i.e., definitions) of two or more target words. The actual senses of the target words are then assumed to be those whose glosses have the greatest word overlap. For example, in the case of two words w_1 and w_2, the Lesk score is defined as $\text{Score}_{Lesk}(S_1, S_2) = |gloss(S_1) \cap gloss(S_2)|$, where $S_1 \in Senses(w_1)$, $S_2 \in Senses(w_2)$ and $gloss(S_i)$ is the bag of words in the definition of sense S_i of w_i. Senses which score the highest value from the above calculation are assigned to the respective words.

While this approach is feasible when the context is small (e.g., two words) it leads to combinatorial explosion as the number of words increases. For example, in a two-word context the number of gloss overlap calculations is $|senses(w_1)| \cdot |senses(w_2)|$, whereas in the case of an n-word context, this increases exponentially to $|senses(w_1)| \cdot |senses(w_2)| \cdot \ldots \cdot |senses(w_n)|$. For this reason, a simplified version of this approach is commonly used, in

which the actual sense for word w is selected as the one whose gloss has the greatest overlap with the words in the context of w. That is, $\text{Score}_{LeskVar}(S) = |context(w) \cap gloss(S)|$, where $context(w)$ is the bag of words in a context window that surrounds the word w.

The method that we propose similarly disambiguates words one at a time; however, rather than using the context provided only in some fixed-size context window surrounding the target word, the method disambiguates the target word using the context provided by all remaining words in the sentence. Essentially, the algorithm computes the semantic similarity (not overlap) between WordNet glosses of the target polysemous word and the text made up of all of the remaining words in the sentence, which we refer to as *context text*. The target word is then assigned the sense associated with the gloss which has the highest semantic similarity score to the context text. This procedure is then repeated for all other words in the sentence.

To formalize, let $W = \{w_i \mid i=1..N\}$ be the set of non-stopwords in the sentence containing the words to be disambiguated, and suppose that we wish to disambiguate word w_i. Let G_{w_i} be the set of WordNet glosses corresponding to word w_i,; i.e., $G_{w_i} = \left\{ g_{w_i}^k \mid k = 1..N_{w_i} \right\}$, where N_{w_i} is the number of WordNet senses for w_i, and $g_{w_i}^k$ is the set of non-stopwords in the k^{th} WordNet gloss of w_i. Let R_i be the context vector comprising all words from W, except w_i; i.e., $R_i = \left\{ w_j \mid w_j \in W, j \neq i \right\}$. The sense for word w_i is identified as the k value for which $g_{w_i}^k$ is semantically most similar to R_i.

The problem, therefore, is again one of calculating the similarity between two text fragments: the gloss, and the context text. The situation is thus somewhat circular, as our motivation for introducing word sense disambiguation was to improve the measurement of short-text similarity. Since attempting to identify the sense of polysemous words in the gloss and context vectors would lead to an infinite regress, we use only the first WordNet sense in comparing these vectors, and define the similarity between a gloss and context vector simply as their cosine similarity in the reduced vector space.

3.1 Word-to-Word Semantic Similarity

Various word-to-word similarity measures have been proposed in the literature, and can broadly be categorized as either *corpus-based*, in which case similarity is calculated based on distributional information derived from large corpora, or *knowledge-based*, in which similarity is based on semantic relations expressed in external resources such as dictionaries or thesauri. In this paper we use knowledge-based measures; specifically, we use lexical knowledge-base WordNet [10].

Two widely used WordNet-based measures, and the measures used in this research, are *shortest path similarity* [12] and the *Jiang and Conrath* [13] measure. Shortest path similarity is the simpler of the two, and is defined as:

$$Sim_{Path}(w_1, w_2) = \frac{1}{length(w_1, w_2)} \tag{1}$$

where *length* is the length of the shortest path between two words, and is determined by simple node counting. The Jiang and Conrath measure [13] is a more sophisticated

measure, based on the idea that the degree to which two words are similar is proportional to the amount of information they share:

$$Sim_{J\&C}(w_1, w_2) = \frac{1}{IC(w_1) + IC(w_2) - 2 \times IC(LCS(w_1, w_2))} \quad (2)$$

where $LCS(w_1, w_2)$ is the word that is the *deepest common ancestor* of w_1 and w_2 in the WordNet hierarchy, $IC(w)$ is the *information content* of word w, and is defined as $IC(w) = -\log P(w)$, where $P(w)$ is the probability that word w appears in a large corpus (e.g., the Brown corpus).

Unlike shortest path, the Jiang and Conrath measure is not capable of calculating the similarity between words with different part-of-speech. For this reason, we use the shortest path measure in the word sense disambiguation phase where we are assuming WordNet sense 0 for words in the gloss and context vectors, and use either of the measures when calculating similarity between sense-assigned words. For a comprehensive review of these and other word similarity measures, see Budanitsky & Hirst (2006) [14].

4 Empirical Results

We present results from applying the similarity measure to three benchmark datasets: the Microsoft Research Paraphrase (MSRP) Corpus [15], the recognizing textual entailment challenge (RTE2, RTE3) [16], and the 30 sentence pairs dataset [17].

4.1 Paraphrase Recognition

The MSRP dataset consists of 5801 pairs of text fragments collected from a large number of web newswire postings over a period of 18 months. Each pair was manually labelled with a binary true or false value by two human annotators, indicating whether or not the two fragments in a pair were considered a paraphrase of each other. The corpus is divided into 4076 training pairs and 1725 test pairs. Since the proposed algorithm is unsupervised (i.e., does not require training from labelled data), we use only test data. Since it is a binary classification task, a classification threshold needs to be determined (i.e., the candidate pair is classified as a paraphrase if the similarity score exceeds this threshold), and the ideal method for comparing performance between classifiers is to look at the performance corresponding to different thresholds; e.g., by comparing area under ROC-curve. Unfortunately this information is not available for other methods, so in line with other researchers, we consider thresholds in increments of 0.1, and provide results corresponding to the best threshold, which in this case was 0.6. Table 1 compares the performance of our measure with other recently reported approaches, as well as two baselines. Our measure achieves best performance in terms of both overall accuracy and F-measure, and far exceeds the baselines. Best performance achieved by a human judge was 83%.

4.2 Textual Entailment Recognition

Textual entailment recognition is the task of determining whether a text fragment is entailed by a hypothesis (another text fragment). Entailment is an asymmetric relation

Table 1. Comparison of performance with other techniques on MSRP classification dataset. Vector-based baseline measures cosine similarity between vectors in a full bag-of-words representation with *tf-idf* weighting. Random baseline was created by randomly assigning a true or false value to pairs of text fragments. Baselines are due to Mihalcea *et al.* (2006) [9].

Measure	Acc	Prec	Rec	F
Proposed Similarity Measure				
J&C	**74.6**	75.5	91.5	**82.7**
Path	**73.2**	73.9	92.4	**82.1**
Islam & Inkpen (2008), Corpus-based				
STS	72.6	74.7	89.1	81.3
Mihalcea *et al.* (2006), Corpus-based				
PMI-IR	69.9	70.2	95.2	81.0
LSA	68.4	69.7	95.2	80.5
Mihalcea *et al.* (2006), WordNet-based				
L&C	69.5	72.4	87.0	79.0
J&C	69.3	72.2	87.1	79.0
Baselines				
Vector-based	65.4	71.6	79.5	75.3
Random	51.3	68.3	50.0	57.8

based on directional inference, and symmetric similarity measures such as that proposed in this paper should not be expected to perform as well as measures designed to utilize a deeper semantic analysis specifically to determine entailment. Nevertheless, the dataset has been previously been used as a measure of (asymmetric) sentence similarity, and we follow suit.

Table 2 shows performance of our measure compared with that recently reported by Ramage *et al.* (2009) [19]. Note that two sets of results are reported in [19]: one set in which the Random Graph Walk method is used as a stand–alone measure, and a second set in which the graph walk method is incorporated within an existing RTE system (i.e., a system designed specifically to detect entailment) [20]. The baseline represents the original performance of this RTE system [20]. The performance of our measure markedly exceeds that of both the baseline and the Ramage *et al.* (2009) measure used a stand-alone. It also performs better on the RTE 3 dataset than the Ramage *et al.* [19] method incorporated into the RTE system, but approximately equally on the RTE2 dataset. As noted above, participants in the RTE challenge have used a variety of strategies beyond lexical relatedness, and accuracies as high as 75.4% [16] and 80% [21] respectively have been reported on the RTE2 and RTE3 datasets.

4.3 30-Sentences Dataset

This dataset is due to Li *et al.* (2006) [16], and was created by taking a set of 65 noun pairs, replacing the nouns with their dictionary definitions, and having 32 human participants rate the similarity in meaning of each of the sentence pairs on a scale of 0.0 to 4.0. When the similarity scores were averaged, the distribution of the scores was heavily skewed toward the low similarity end of the scale, with 46 pairs rated from 0.0 to 0.9, and 19 pairs rated from 1.0 to 4.0. To obtain a more even distribution across the similarity range, a subset of 30 sentence pairs was selected, consisting of all 19 sentence

pairs rated 1.0 to 4.0, and 11 taken at equally spaced intervals from the 46 pairs rated 0.0 to 0.9, [17]. Unlike the dataset described above, in which the task is binary classification, this dataset has been used to compare correlation with human-rated similarity.

The similarity measures proposed in Islam and Inkpen (2008) [18] and Li *et al.* (2006) [8] achieved correlations of 0.853 and 0.816 respectively on this task. Our sentence similarity measure exceeds both of these results, achieving correlations of 0.877 and 0.874 respectively using J&C and Path Length word-to-word similarity measures. These figures also exceed the mean human correlation of 0.825, and are not far from the highest correlation of 0.921 achieved by a human participant

Table 2. Comparison of performance against results recently reported by Ramage et al. (2009) [19], and a baseline RTE system [20]. Classification threshold is 0.5.

Measure	RTE3 Accuracy	RTE2 Accuracy
Proposed similarity measure		
Path	**70.2**	**62.8**
J&C	**68.7**	**63.8**
Ramage *et al*., (2009) with Random Graph Walk		
Cosine	55.7	57.0
Dice	55.7	54.2
Jensen-Shannon	56.7	57.5
Ramage *et al*., (2009) with existing RTE system		
Cosine	65.8	**64.5**
Dice	65.4	63.1
Jensen-Shannon	65.4	63.2
Baselines		
Existing RTE3	65.4	63.6

5 Conclusion

The results from the previous section are positive, and suggest that incorporating word sense disambiguation and synonym expansion does to lead to improvement in sentence similarity measurement. Importantly, this improvement is gained with very little increase in computational cost. Although we have described in this paper how these ideas can be incorporated into a measure based on a reduced vector space representation, the ideas can readily be applied to measures such as that of Mihalcea *et al.* (2006), which do not use an explicit vector space representation. While the empirical results reported in this paper have focused mainly on binary classification tasks, we believe that an important test for a sentence similarity measure is how well it performs when used in the context of a more encompassing task such as text mining or document summarization. Evaluating sentence similarity measures in such contexts is difficult, however, as many different factors play a role in the success of such systems, and it is difficult to isolate the effect that any specific sentence similarity measure may have. As a step towards such a broader evaluation, we are in the process of applying the measure to challenging sentence clustering tasks. We are also comparing the performance of the disambiguation algorithm with that of other disambiguation algorithms directly on standard WSD datasets.

References

1. Atkinson-Abutridy, J., Mellish, C., Aitken, S.: Combining Information Extraction with Genetic Algorithms for Text Mining. IEEE Intelligent Systems 19(3), 22–30 (2004)
2. Erkan, G., Radev, D.: LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. Journal of Art. Int. Research. 22, 457–479 (2004)
3. Bilotti, M.W., Ogilvie, P., Callan, J., Nyberg, E.: Structured Retrieval for Question Answering. In: SIGIR 2007, pp. 351–358. ACM, New York (2007)
4. Mohler, M., Mihalcea, R.: Text-to-Text Semantic Similarity for Automatic Short Answer Grading. In: EC-ACL 2009, Athens, Greece, pp. 567–575 (2009)
5. Coelho, T., Calado, P., Souza, L., Ribeiro-Neto, B., Muntz, R.: Image Retrieval using Multiple Evidence Ranking. IEEE Tran. On KDD 16(4), 408–417 (2004)
6. Salton, G.: Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley, Reading (1989)
7. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
8. Li, Y., McLean, D., Bandar, Z., O'Shea, F., Crockett, K.: Sentence Similarity Based on Semantic Nets and Corpus Statistics. IEEE TKDE 18(8), 1138–1150 (2006)
9. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and Knowledge-based Measures of Text Semantic Similarity. In: 21st National Conference on Art. Int., vol. 1, pp. 775–780 (2006)
10. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge
11. Lesk, M.: Automatic Sense Disambiguation using Machine Readable Dictionaries: How to tell a pine cone from an ice cream cone. In: Proc. of the SIGDOC, pp. 24–26 (1986)
12. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and Application of a Metric to Semantic Nets. IEEE Trans. Sys., Man and Cyb. 19(1), 17–30 (1989)
13. Jiang, J.J., Conrath, D.W.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: 10th Inter. Conf. on Research in Computational Linguistics, pp. 19–33 (1997)
14. Budanitsky, A., Hirst, G.: Evaluating WordNet-based Measures of Lexical Semantic Relatedness. Computational Linguistics 32(1), 13–47 (2006)
15. Dolan, W., Chris Quirk, C., Brockett, C.V.: Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In: 20th International Conf. on Computational Linguistics, pp. 350–356 (2004)
16. Dagan, I., Dolan, B., Giampiccolo, D., Magnini, B.: The Third PASCAL Recognizing Textual Entailment Challenge. In: ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 1–9 (2007)
17. Li, Y., McLean, D., Bandar, Z., O'Shea, F., Crockett, K.: Pilot Short Text Semantic Similarity Benchmark Data Set: Full Listing and Description (2009), `http://www.mendeley.com`
18. Islam, A., Inkpen, D.: Semantic Text Similarity using Corpus-based Word Similarity and String Similarity. ACM Trans. on KDD 2(2), 1–25 (2008)
19. Ramage, D., Rafferty, A., Manning, C.: Random Walks for Text Semantic Similarity. In: ACL-IJCNLP 2009, pp. 23–31 (2009)
20. Chambers, N., Cer, D., Grenager, T., Hall, D., Kiddon, C., MacCartney, B., de Marneffe, M., Ramage, D., Yeh, E., Manning, C.: Learning Alignments and Leveraging Natural Logic. In: ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 165–170 (2007)
21. Hickl, A., Bensley, J.: A Discourse Commitment-based Framework for Recognizing Textual Entailment. In: ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 171–176 (2007)

LEXA: Towards Automatic Legal Citation Classification

Filippo Galgani and Achim Hoffmann

School of Computer Science and Engineering
The University of New South Wales, Sydney, Australia
{galganif,achim}@cse.unsw.edu.au

Abstract. In this paper we present our approach towards legal citation classification using incremental knowledge acquisition. This forms a part of our more ambitious goal of automatic legal text summarization. We created a large training and test corpus from court decision reports in Australia. We showed that, within less than a week, it is possible to develop a good quality knowledge base which considerably outperforms a baseline Machine Learning approach. We note that the problem of legal citation classification allows the use of Machine Learning as classified training data is available. For other subproblems of legal text summarization this is unlikely to be the case.

1 Introduction

The legal field is strongly affected by the problem of information overload, due to the large amount of legal material stored in textual form. Past decisions can have a binding effect on following decisions, in a process that is known as *stare decisis* [12], especially in countries with common law systems, such as Australia, UK and USA. As a consequence, judges need to know past cases to be coherent and "just" in their application of law and lawyers use them to find arguments for their cases. Court decisions or cases can be instructive as they introduce a new principle or rule, modify or interpret an existing principle or rule, or settle a question upon which the law is doubtful.

A number of different approaches of information management from other domains have been carried over to the legal domain: for example automatic summarization [9,6], retrieval [11] and information extraction [16]. However, researchers in the field, such as Hachey and Grover, already noted:

> "[...] *the legal domain appears to be more complex than scientific articles and especially news, the most commonly reported domains in the automatic summarization literature. This is evidenced in characteristics of legal discourse such as the longer average sentence lengths, longer average document lengths, and the sometimes convoluted and philosophical nature of legalese where there is not an absolute logical template and there is a looser notion of topic which lends itself to a less centralized focus*" [8].

J. Li (Ed.): AI 2010, LNAI 6464, pp. 445–454, 2010.

Automatic summarization of legal cases can support finding (helping in assessing the relevance of the results of a query) and digesting the right documents. Furthermore it can aid the manual creation of summaries and provide important legal information in a format that is more accessible and understandable. In case-law systems, such as in Australia, because of the importance of relying on other cases to answer the case at hand, citations are an important aspect of most judicial decisions. Examining citations tells us how the law we are relying on has been interpreted. For this reason it is vital to law professionals to know whether the decision has received positive, negative, cautionary or neutral treatment in subsequent judgements.

In this paper we outline our system LEXA (Legal tEXt Analyzer): an approach towards automatically providing information useful to law professionals from such case reports. LEXA is based on incremental acquisition of annotation rules, and we describe an evaluation of it on a citation classification task.

The following section discusses related work. This is followed by the description of our annotated legal corpus in Section 3. In Section 4, we present our approach towards legal citation classification. The following Section 5 discusses our first results on legal citation classification. The final Section 6 discusses our achievements so far and outlines future research.

2 Related and Prior Work

In the past automatic summarization has attracted a large body of research, and a large variety of techniques and approaches have been proposed for this task. Although there has been a certain amount of research in summarization of legal texts, this application domain is not mature as other such as news or scientific articles. Examples of systems for automatic summarization of legal texts are LETSUM [6] and the work of Hachey and Grover [9].

To our knowledge there have been no attempt to automatically classify citations in legal cases, citation classification has been applied mainly in the domain of scientific papers. Following the pioneering approach of Nanba and Okomura [15], another system that automatically perform citation classification is described in [20], where different kinds of features are studied to train an IBk classifier on an annotated corpora.

In 2004 Nakov et.al. [14], pointed out the possibility to use citations contexts directly for text summarization, as they provide information on the important facts contained in the paper. A first application of the idea can be found in the work of Qazvinian and Radev [18], where they create a summary by extracting a subset of the sentences that constitute the citation context. Mohammad et.al. [13] apply this approach to multi-document summarization, they also build up on the claim by Elkiss et.al. [5] about the difference of information given by the abstract and the citation summary of a paper.

Our knowledge acquisition (KA) methodology is based on incremental approaches, in particular on the Ripple Down Rules (RDR) methodology [3]. In RDR, rules are created manually by domain experts without a knowledge engineer. The knowledge base is built with incremental refinements from scratch,

while the system is running: a domain expert monitors the system and whenever it performs incorrectly, he signals the error and provides a rule as a correction.

A Single Classification RDR (SCRDR), see Figure 1, is a binary tree; associated with each node is a rule (a condition and a conclusion). Cases (objects to be classified) are evaluated as they are passed from node to node, starting from the root: if the condition of the node is satisfied we follow the so-called *except* edge (we say that the node *fires*), otherwise the *if not* edge, if there is any. The final conclusion given by the SCRDR tree is the conclusion of the node that fired last, i.e. that is deepest in the tree (but is often not a leaf node). To ensure that a conclusion is always found, the root node typically contains a trivial condition which is always satisfied. This node is called the *default* node. When an instance is misclassified, a new node is added to the tree. If the node n_f that fired last has no except link, a new except link is created and the new node is attached to it. If n_f has already an except link leading to node n_e the new node is attached as an alternative except link. That is done by following the *if-not* link chain of n_e until no *if-not* link is found and then creating a new *if-not* link and attaching the new node to that link. Then the domain expert formulates a rule for the new node that is satisfied by the case. This rule represents an explanation for why the conclusion on the case at hand should be different. The strength of RDR is easy maintenance: the point of failure is automatically identified, the expert patches the knowledge only locally, considering the case at hand, and new rules are placed by the system in the correct position and checked for consistency with all cases previously correctly classified.

RDR have been applied to different problems and applications. For a recent survey see [19]. RDR has also been extended to tackle natural language processing tasks. Among such work is also the work on scientific citation classification in [17] on which the work in this paper builds to a significant extent. However, our application domain is considerably more complex. Hence, this paper demonstrates that the Ripple Down Rules approach also successfully extends to more complex NLP domains.

3 Creating Our Corpus of Legal Citations

AustLII (the Australasian Legal Information Institute) [1,7] provides free access to a large amount of legal information, including reports on court decisions in all major courts in Australia. A similar project, the World Legal Information Institute (WordLII), is an extension to other countries, with the aim of providing "*free, independent and non-profit access to worldwide law*".

We accessed the court case reports in html format from the AustLII website. Some of the contained citations are marked up with a hyperlink to the corresponding cited case. Notably, some of the Federal Court of Australia (FCA) cases also contain expert generated citation classes of the cited cases.

We built a robust parser to analyse the html pages of the FCA reports and extract the relevant information about the citations.

Examples of the classified citations in FCA documents are:

- Dunstan v Human Rights and Equal Opportunity Commission (No 2) [2005] FCA 1885 related
- Australian Fisheries Management Authority v PW Adams Pty Ltd (No 2) (1996) 66 FCR 349 distinguished
- Copping v ANZ McCaughan Ltd (1997) 67 SASR 525 cited
- DJL v Central Authority [2000] HCA 17; (2000) 201 CLR 226 considered

We can decompose each row in the name (e.g. DJL v Central Authority), the legal citations (e.g. [2000] HCA 17) and the class (e.g. considered). The distribution among the citation classes for the years 2007-2009 from 2043 FCA documents containing 18715 labelled citations is shown in Table 1.

Table 1. Distribution of citation classes for 2007-2009. Those selected for our knowledge acquisition task are in bold.

Cited	9346	Referred to	3017	**Applied**	**1803**	**Followed**	**1759**
Considered	1339	Discussed	706	**Distinguished**	**463**	Related	94
Affirmed	91	Quoted	87	Approved	61	Not Followed	57
Reversed	20	Ref to	15	Explained	10	Questioned	9
Disapproved	8	Noted	7	Relied on	4	Doubted	3
Compared	2	Adopted	2	Overruled	2	Referred	2

It is possible that a case is cited differently, in different citing cases, or even within the same citing case, due to the fact that different aspects of the cited case may be of interest. As a consequence, combinations such as Applied/Distinguished are possible and one citation may have more than one class label attached, though this is rare.

Finding all occurrences of a citation in a case is not trivial, as references are made in a large variety of ways as opposed to scientific articles. Different ways of referring to the same case include:

- The full name of the case, e.g. Yevad Products Pty Ltd v Brookfield [2005] FCAFC 263; (2005) 147 FCR 282, or just Yevad Products Pty Ltd v Brookfield
- The name of one of the parts, e.g. Yevad or Brookfield
- Indication of the law report, e.g. (2005) 147 FCR 282, or the medium neutral citation, e.g. [2005] FCAFC 263
- Combination of these components, for example Brookfield 147 FCR
- The name of the respective judge, e.g. *In Burgundy Royale Brennan J at 685 said:* or *I understand Brennan J's reference to the prospect of a grant of special leave...*

When resolving the last type of reference we only use those citations that are unambiguous, e.g. where a judge's name is only involved in one possible cited case. To turn this information into a training corpus usable for supervised machine

learning, we attached to each citation its class label and the associated sentence(s) as well as the entire paragraph(s) in which the citation occurs. Where a citation is mentioned multiple times we collect the sentences (paragraphs) surrounding each of the occurrences. For our study we only used a subset of our training corpus as described in Section 5, but we are going to make the entire corpus available for other researchers.

4 Building Legal Citation Classification Systems

Our citation classification system is based on creating a knowledge base of rules to annotate text, and in particular to annotate citations. We built an application based on GATE [4]. GATE is a framework for developing components for processing human language including text, written in Java and available as free open-source software. We use the Tokenizer, Sentence Splitter, Part of Speech Tagger and Stemmer resources (provided with GATE) to generate Token annotations and their corresponding features for input texts, building the first layer of linguistic annotations.

Our system LEXA for classification of citations is based on customized annotations produced by a knowledge base of rules, which take the form of regular expressions over annotations. Each rule matches a regular expression of tokens and other annotations, and posts a new annotation. To create these annotations we used the Semantic Tagger from GATE, a finite state transducer which annotates text based on JAPE (Java Annotation Patterns Engine) grammars.

Our system is composed of different types of rules. A first group of rules is used in a preliminary phase to identify judge names, parts (e.g. "the plaintiff", "the appellant"...), courts (e.g. "a full court of the HCA"...), citations of paragraphs (e.g. "case at [145-148]"), etc. The user can create any type of annotation as needed. A successive level of annotation aims at posting class labels over citations. This class of rules posts a particular annotation which specify which class we believe the citation belongs to.

This set of classification rules is acquired interacting with a human user. The system displays citations (and surrounding text) to the user, specifying the (known) type of citation. The user, examining the text at hand, can create a new rule and test it on the entire corpus. At the first level we created rules to extract Distinguished citations. The user is presented with the surrounding text of a Distinguished case not yet classified, and creates a rule to identify that case. The rule can be tested on the entire corpus to see how many Distinguished and Followed/Applied cases it matches. When the user is satisfied with the rule, he/she commits the rule to the knowledge base.

When a new case of class Followed/Applied is considered and incorrectly classified as Distinguished by a first level rule, the user will add a second level exception rule reverting the tentative classification of Distinguished to Followed/Applied.

After the knowledge base is built, the classification of new citation cases is done as described in Section 2. An example of a portion of the resulting RDR tree, including three rules, is shown in Figure 1. We conducted a number of

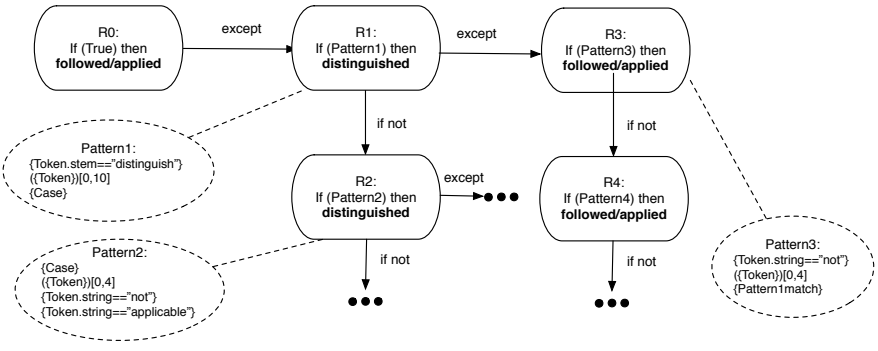


Fig. 1. Example of a portion of the RDR tree: R1 matches a word whose stem is "distinguish", followed by an annotation of type Case (the annotation that signal the cases in our corpus), separated by a gap up to 10 words. The exception rule R3 looks if there is a token "not" up to 4 tokens before the annotation posted by R1.

knowledge acquisition sessions (no legal expert was involved) in order to build a knowledge base with a reasonable performance on our legal citation classification corpus.

5 Experimental Results

We considered a two class problem: Distinguished (D) vs. Followed or Applied (FA). We believe that these classes are particularly relevant for legal professionals as they are likely to shed light on what criteria have been used to decide if a given case constitutes a relevant precedent (for Followed or Applied). Similarly, Distinguished cases would indicate why a cited case is not relevant for the citing case. The importance of these three classes is further corroborated by the fact that alternative classification schemes, e.g. from private publishers such as Lexis Nexis, coincide on those classes while they differ on many other classes. The Lexis Nexis CaseBase Case Citator [2] explains these classes as follows:

- Applied: A principle of law articulated in the primary case is applied to a new set of facts by the court in the subsequent case.
- Followed: The annotation is similar to applied but is used in circumstances where the facts in the primary case resemble reasonably closely the facts in the subsequent consideration case.
- Distinguished: The court in the subsequent case holds that the legal principles articulated by the primary case (usually otherwise persuasive or binding authority) do not apply because of some difference between the two cases in fact or law.

The respective sub-corpus for the three classes we used contains 460 Distinguished citations and 3496 Followed or Applied citations with a total of 3956 citations.

Using the approach sketched in Section 4, we developed a first knowledge base taking around 30 hours of knowledge acquisition sessions. The knowledge base contains a total of 34 rules of which 15 are second-level exception rules. Only seven of the first layer rules contain one (or more) exception rules. The time to acquire a rule is divided into reading and understanding the text and time to decide which rule should be created. The first is the most demanding part, as sometimes the text is long and complex. Sometimes it is difficult to understand why a certain class label was given.

Comparing Our KA Approach with Machine Learning

While developing our knowledge base, we could test the performance of the rules directly on the corpus. The root node of the RDR tree classify every case as FA, i.e. it misclassifies 460 D cases. After adding exceptions rules to the default node, we recognized 207 citations of class D correctly. However also 133 cases were incorrectly classified as D. In order to rectify that we added exception rules to the first level of exception rules classifying cases as D. Those second level exception rules recognize correctly 80 out of the 133 FA cases. This second level of exceptions rules also caused 15 out of the 207 D cases to be misclassified as FA.

The knowledge base we developed in some 30 hours of knowledge acquisition. To create the 34 rules, we looked at approximately 60 cases (but we tested them on all cases before deciding to commit a formulated rule). Some of the rules were dismissed after testing and then manually refined.

To compare the performance of our knowledge base with a baseline machine learner we trained the Naive Bayes classifier in WEKA [10] using a simple bag of words model, only indicating presence or absence of a word. To build the model, for each citation we extract all the words that appear in the surrounding context (either sentence or paragraph). Comparison with alternative machine learning approaches is left for future research.

Due to the fact that NB did not do well on recognizing the minority class of Distinguished cases (which we believe is the more important one for legal practitioners) we tried to improve the Naive Bayes performance by giving more weight to the minority class, i.e. by replicating the instances of type D, with factors of 2, 4, 8, 12 and 20. We found the factors 2 and 4 to produce the best results. The results are presented in Table 2. Six Naive Bayes models are shown, using only the words present in the same sentence (s) or all the words in the paragraph (p), with the original instances or multiplying factor of 2 and 4.

We built a test corpus of unseen data by downloading all FCA cases for 2006 containing a total of 6541 citations with 1274 FA and 160 D citations. This data had never been used at any stage in developing our system. We applied our knowledge base as well as the six trained Naive Bayes classifiers to the new test data. The results are shown in Table 2.

It should be noted, however, that the human expert provided class labels are not necessarily agreed upon by other human experts. This results effectively in noise in our data for the purpose of training and testing our classifiers.

Table 2. NB2s: stands for Naive Bayes with words from surrounding sentence supplied and cases from class D provided twice - other column titles analogously

	Training data 2007-2009							Test data 2006						
	LEXA	NBs	NB2s	NB4s	NBp	NB2p	NB4p	LEXA	NBs	NB2s	NB4s	NBp	NB2p	NB4p
Precision(D)	0.784	0.632	0.814	0.824	0.806	0.784	0.208	0.5	0.137	0.303	0.493	0.219	0.359	0.167
Recall (D)	0.417	0.25	0.485	0.772	0.172	0.898	1	0.263	0.1	0.188	0.206	0.088	0.144	0.606
F-measure(D)	0.545	0.358	0.608	0.797	0.283	0.837	0.344	0.344	0.116	0.232	0.291	0.125	0.205	0.262
Precision(FA)	0.928	0.91	0.937	0.971	0.903	0.987	1	0.913	0.891	0.903	0.907	0.894	0.9	0.926
Recall (FA)	0.985	0.981	0.986	0.979	0.995	0.968	0.506	0.967	0.921	0.946	0.973	0.961	0.968	0.622
F-measure(FA)	0.955	0.944	0.96	0.975	0.946	0.977	0.672	0.939	0.906	0.924	0.939	0.926	0.933	0.744
Accuracy	0.919	0.897	0.928	0.955	0.9	0.96	0.562	0.888	0.829	0.861	0.888	0.864	0.876	0.62

Table 3. Removing cases with differing human expert classifications

	Training data 2007–2009							Test data 2006						
	LEXA	NBs	NB2s	NB4s	NBp	NB2p	NB4p	LEXA	NBs	NB2s	NB4s	NBp	NB2p	NB4p
Precision(D)	0.8	0.545	0.804	0.911	0.875	0.912	0.463	0.674	0.154	0.341	0.563	0.214	0.64	0.362
Recall (D)	0.516	0.293	0.511	0.684	0.187	0.827	0.987	0.403	0.083	0.194	0.25	0.083	0.222	0.694
F-measure(D)	0.627	0.382	0.625	0.782	0.308	0.867	0.63	0.504	0.108	0.248	0.346	0.12	0.33	0.476
Precision(FA)	0.903	0.863	0.904	0.936	0.852	0.964	0.996	0.850	0.772	0.799	0.818	0.781	0.816	0.885
Recall (FA)	0.972	0.948	0.974	0.986	0.994	0.983	0.757	0.946	0.872	0.895	0.946	0.914	0.965	0.658
F-measure(FA)	0.936	0.904	0.937	0.96	0.918	0.973	0.86	0.895	0.819	0.844	0.877	0.842	0.884	0.754
Accuracy	0.891	0.833	0.893	0.933	0.853	0.956	0.797	0.827	0.699	0.742	0.793	0.732	0.802	0.666

To identify those "noisy" or at least questionable citations, we compared the FCA citation classifications with the available classifications of the same citations by the Lexis Nexis CaseBase Case Citator [2], a commercial database of case law. In this database experts classify each citation in a scheme of eleven classes (different from the one of FCA but comparable to it). Of our 460 FCA cases marked as Distinguished, 225 of them had the same label in CaseBase. Of the 3496 Followed or Applied, only 1041 received either Followed or Applied in CaseBase.

As a consequence of this considerable discrepancy in human expert opinion, which we think is due to the class boundaries not being very sharp, it appears to be more appropriate to limit the performance evaluation in our study to those cases where both human-provided class labels (from FCA and Lexis Nexis CaseBase) agree. Results for the training and test sets (329 citations) containing only the cases the human experts agreed on, are shown in Table 3.

These results indicate that the human intuition that went into the knowledge base of our system LEXA generalises significantly better than the Machine Learner which appears to be overfitting the training data to a much higher degree than LEXA (the performance difference between training and test data is less for LEXA), with our system obtaining an accuracy of 82.7%. As a consequence, on the less ambiguous sub-corpus our knowledge base is outperforming the best version of our trained Naive Bayes classifier by a margin of up to some 45% relative to the Naive Bayes F-measure of 34.6% for the important D class.

6 Conclusions and Future Work

In this paper we presented our approach towards automatic legal citation classification, which characterizes the relation between the present case and the cited ones. Automatic classification of citations to case law is a novel application in itself, which we believe to be very relevant in assisting legal research in common

law. Moreover, we believe that citation analysis in legal cases can bring benefits to a range of other NLP applications, including automatic summarization. We built an annotated corpus of citations, extracting available data from the Federal Court of Australia reports, which we are going to release to interested researchers.

Our system LEXA is based on a knowledge base of rules (described by regular expressions) to annotate text at multiple levels. We built a knowledge base manually acquiring a set of rules: within less than a week of knowledge acquisition sessions, our system is able to recognize Distinguished and Followed or Applied citations with an accuracy of 88.8% on test data. When evaluated on unseen data, the system outperforms our best Machine Learning model, giving significantly higher recall and precision for the Distinguished class (34.4% vs 29.1% F-measure).

Examining an alternative source of classification of the same cases, we found that human experts often disagree when classifying citations (only 1266 of the 3956 citations have the same label in both sources). This confirms the complexity of citation classification in the legal domain. We experimented that, when taking out "ambiguous" citations (the ones on which human experts do not agree), the performance increases for our knowledge base, with our system bringing a considerable improvement of F-measure both for Distinguished (50.4% vs. 34.6%) and Followed/Applied (89.5% vs. 87.7%) over Naive Bayes.

Future work involves integrating this approach in a more comprehensive analysis of the legal texts, with the aim to build an automatic summarization system. In order to achieve this, more rules of different types will be needed, to allow deeper analysis of the text. For automatic summarization the problems in obtaining annotated data makes it more difficult to use Machine Learning approaches. For this reason we think that knowledge acquisition from experts with RDR is the path to follow to obtain better results.

References

1. Australasian Legal Information Institute, http://www.austlii.edu.au/
2. Lexis Nexis CaseBase Case Citator, http://www.lexisnexis.com.au/
3. Compton, P., Jansen, R.: Knowledge in context: a strategy for expert system maintenance. In: AI 1988: Proceedings of the second Australian Joint Conference on Artificial Intelligence, pp. 292–306. Springer, New York (1990)
4. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: Gate: A framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL 2002), Philadelphia (July 2002)
5. Elkiss, A., Shen, S., Fader, A., Erkan, G., States, D., Radev, D.: Blind men and elephants: What do citation summaries tell us about a research article? J. Am. Soc. Inf. Sci. Technol. 59(1), 51–62 (2008)
6. Farzindar, A., Lapalme, G.: Letsum, an automatic legal text summarizing system. In: Legal Knowledge and Information Systems: JURIX 2004, the Seventeenth Annual Conference, p. 11. Ios Pr. Inc., Amsterdam (2004)

7. Greenleaf, G., Mowbray, A., King, G., Van Dijk, P.: Public Access to Law via Internet: The Australian Legal Information Institute. Journal of Law and Information Science 6, 49 (1995)
8. Hachey, B., Grover, C.: Automatic legal text summarisation: experiments with summary structuring. In: ICAIL 2005: Proceedings of the 10th International Conference on Artificial Intelligence and law, pp. 75–84. ACM, New York (2005)
9. Hachey, B., Grover, C.: Extractive summarisation of legal texts. Artif. Intell. Law 14(4), 305–345 (2006)
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. 11(1), 10–18 (2009)
11. Moens, M.F.: Innovative techniques for legal text retrieval. Artificial Intelligence and Law 9(1), 29–57 (2001)
12. Moens, M.F.: Summarizing court decisions. Inf. Process. Manage. 43(6), 1748–1764 (2007)
13. Mohammad, S., Dorr, B., Egan, M., Hassan, A., Muthukrishan, P., Qazvinian, V., Radev, D., Zajic, D.: Using citations to generate surveys of scientific paradigms. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Boulder, Colorado, pp. 584–592 (June 2009)
14. Nakov, P.I., Schwartz, A.S., Hearst, M.A.: Citances: Citation sentences for semantic analysis of bioscience text. In: Proceedings of the SIGIR 2004 Workshop on Search and Discovery in Bioinformatics (2004)
15. Nanba, H., Okumura, M.: Towards multi-paper summarization using reference information. In: IJCAI 1999: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, pp. 926–931. Morgan Kaufmann Publishers Inc., San Francisco (1999)
16. Palau, R.M., Moens, M.F.: Argumentation mining: the detection, classification and structure of arguments in text. In: ICAIL 2009: Proceedings of the 12th International Conference on Artificial Intelligence and Law, pp. 98–107. ACM, New York (2009)
17. Pham, S.B., Hoffmann, A.: A new approach for scientific citation classification using cue phrases. In: Proceedings of Australian Joint Conference in Artificial Intelligence (2003)
18. Qazvinian, V., Radev, D.R.: Scientific Paper Summarization Using Citation Summary Networks. In: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pp. 689–696 (2008)
19. Richards, D.: Two decades of Ripple Down Rules research. Knowl. Eng. Rev. 24(2), 159–184 (2009)
20. Teufel, S., Siddharthan, A., Tidhar, D.: Automatic classification of citation function. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pp. 103–110. Association for Computational Linguistics, Sydney (July 2006)

Hazardous Document Detection Based on Dependency Relations and Thesaurus

Kazushi Ikeda, Tadashi Yanagihara, Gen Hattori,
Kazunori Matsumoto, and Yasuhiro Takisima

KDDI R&D Laboratories Inc., 2–1–15 Ohara, Fujimino, Saitama, 356–8502 Japan
{kz-ikeda,td-yanagihara,matsu,takisima}@kddilabs.jp

Abstract. In this paper, we propose algorithms to increase the accuracy of hazardous Web page detection by correcting the detection errors of typical keyword-based algorithms based on the dependency relations between the hazardous keywords and their neighboring segments. Most typical text-based filtering systems ignore the context where the hazardous keywords appear. Our algorithms automatically obtain segment pairs that are in dependency relations and appear to characterize hazardous documents. In addition, we also propose a practical approach to expanding segment pairs with a thesaurus. Experiments with a large number of Web pages show that our algorithms increase the detection F value by 7.3% compared to the conventional algorithms.

Keywords: Information Filtering, Dependency Relation, Thesaurus.

1 Introduction

As Internet use becomes more widespread, the number of Web pages for consumers is increasing, and includes many blogs and bulletin board systems (BBS). Web pages that contain hazardous information on dating, criminal declarations, and libelous statements are also increasing. The labor cost associated with monitoring such Web pages in order to remove them is also becoming considerable. To detect hazardous Web pages, text-based approaches have been proposed that automatically generate a set of hazardous keywords from a training dataset which are manually labeled as hazardous/harmless. However, conventional text-based approaches lack adequate accuracy because they ignore the context where the hazardous keywords appear. For example, the word “kill” is used in both hazardous documents such as “kill a man” and harmless documents such as “kill a process”.

In this paper, we propose algorithms to increase the accuracy of hazardous Web page detection by correcting the classification of the conventional text-based algorithms based on the dependency relations of the hazardous keywords and their neighboring segments. In addition, we propose practical algorithms to increase performance by expanding the hazardous segment pairs using a thesaurus. Currently we are targeting Japanese Web pages; however, the essential algorithms of hazardous document detection are applicable to other languages by alternating language specific parts such as dependency analysis.

J. Li (Ed.): AI 2010, LNAI 6464, pp. 455–465, 2010.

In our experiments, we conducted a large-scale performance evaluation using 220,000 manually labeled Japanese Web pages as training data for our algorithms, and another 20,000 Web pages for the evaluation itself. The performance of our algorithms exceeded that of conventional algorithms by up to 7.3% in F value.

2 Related Works

Several text-based algorithms have been proposed to detect hazardous Web pages [1], [2]. The algorithms proposed in [1] automatically generate a set of hazardous keywords that appear unusually often in hazardous Web pages in the training datasets. The algorithms proposed in [2] calculate the hazard score of Web pages based on the similarity of the feature vectors extracted from Web pages of the training datasets and the evaluation datasets. In these algorithms, however, documents are split into morphemes thus the contexts where they appear are ignored. As a result, they have difficulty in accurately discriminating between documents that contain morphemes that are classified both as hazardous and harmless depending on the context, such as "kill" and "drugs".

Accurate document retrieval algorithms based on dependency relations have been proposed [3], [4]. The algorithms proposed in [3] deal with Japanese documents. The algorithms split documents into morphemes, analyze their dependency relations, and make a binary tree called a "Structured Index" in advance. Users' queries are written in pseudo-natural language, and are also analyzed into the Structured Index and matched with the documents. The algorithms described in [4] expand users' query words by extracting contextual terms and relations from external documents. These algorithms aim at improving the accuracy of document retrieval by using the dependency relations of morphemes. Although the aims of our algorithms are different, focusing on dependency relations to detect hazardous documents with high accuracy is a promising approach.

In addition, term expansion using a thesaurus is also a promising approach to improving performance. Query expansion algorithms are well researched and many types of algorithms have been proposed [5], [6], [7], [8]. Liu's group [5] offers methods to improve query expansion for ambiguous words. In [6], they present an approach to combining WordNet and ConceptNet by assigning appropriate weights for expanded terms. Yoshioka's group [7] proposes algorithms to modify a given Boolean query by using information from a relevant document set by combining probabilistic and Boolean IR models. In the recent research in [8], query expansion algorithms based on users' browsing histories are proposed. In their algorithms, Web pages are clustered into a Web community and each query is represented by the Web communities to which its accessed Web pages belong.

A part of our contribution has been reported in [9]. Going beyond this achievement, we offer practical query expansion algorithms with a thesaurus that improve the detection accuracy for hazardous documents and reduce the computation load and memory consumption by removing noisy segment pairs.

3 Proposed Algorithms

In this paper, we treat the algorithms of [1] as conventional text-based algorithms. Figure 1 shows an overview of the conventional and the proposed algorithms.

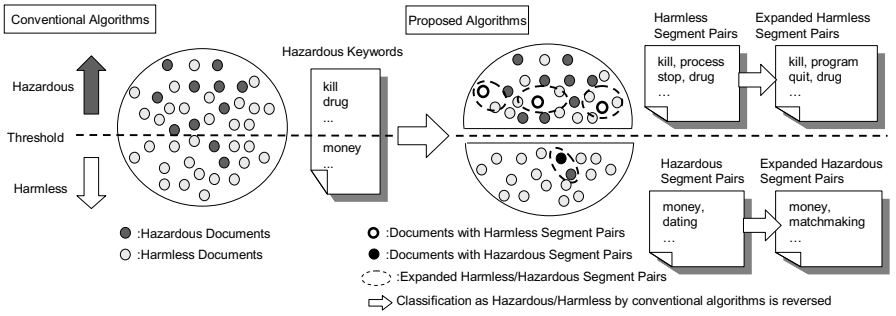


Fig. 1. Overview of the Conventional Algorithms and the Proposed Algorithms

Documents are classified into either hazardous or harmless. The conventional algorithms have an automatically generated set of keywords with a hazard score.

Documents that contain any keywords with a higher score than a threshold are detected as hazardous and the rest are considered harmless. In Figure 1, for example, the words "kill" and "drug" are regarded as hazardous because their hazard scores are higher than the threshold. In contrast, words with low hazard scores such as "money" are regarded as harmless.

Although documents that contain keywords with high hazard scores are likely to be hazardous, not all of them are hazardous. For example, documents which contain sentences such as "kill a process" are not hazardous even though they contain the hazardous keyword "kill". In the same way, documents that contain sentences such as "make money with dating" are hazardous even though the keyword "money" is harmless. Our algorithms correct errors and improve accuracy by detecting hazardous/harmless segment pairs from documents classified as harmless/hazardous, respectively. In addition, we propose algorithms to expand segment pairs with a thesaurus. For example, in our algorithms, the keyword "kill" is harmless when it appears as a segment pair with "process". In addition, expanded segment pairs such as "kill program" and "kill computation" are also regarded as harmless.

In the following section, we describe the conventional generation algorithms of the hazardous keyword set in Section 3.1. We describe the proposed generation algorithms for hazardous/harmless segment pairs in Section 3.2. Expansion algorithms for generated segment pairs are shown in Section 3.3.

3.1 Generation of Keyword Set

First we describe the conventional generation algorithms for a hazardous keyword set. Algorithms shown in [1] split the documents manually labeled as hazardous/harmless into morphemes by morphological analysis and extract hazardous morphemes that appear particularly often in hazardous documents. E(m) which is the degree of bias of a morpheme m in hazardous documents is calculated based on Akaike's Information Criterion (AIC) [10]. In AIC algorithms, the four criteria shown in Table 1 are used, where N_{11}/N_{21} is the number of hazardous/harmless documents where morpheme m appears, N_{12}/N_{22} is the number of hazardous/harmless documents where morpheme m does not appear. In [1], E(m) is defined as follows by using AIC dependent/ independent models of AIC_DM/AIC_IM based on the findings described in [11].

Table 1. Number of Hazardous/Harmless Documents where Morpheme m Appears

	Documents do contain m	Documents do not contain m	Sum
Hazardous Documents	$N_{11}(m)$	$N_{12}(m)$	N_p
Harmless Documents	$N_{21}(m)$	$N_{22}(m)$	N_n
Sum	N(m)	$N(\neg m)$	N

Table 2. Example of the Obtained Hazardous Keywords

Ranks	Keywords	$N_{11}(m)$	$N_{12}(m)$	$N_{21}(m)$	$N_{22}(m)$	E(m)
10	Actress	5802	102724	194	10833	6746
17	Blog	1091	97615	3354	10517	4495
46	Mobile phone	9253	99273	3259	10526	3167
106	Sponsor	2561	105965	708	10781	1129
110	Access	6573	101953	3361	10516	1105

$$\begin{aligned} &When\ \ N_{11}(m)/N(m) \geq N_{12}(m)/N(\neg m) \\ &\qquad E(m) = AIC\_IM(m) - AIC\_DM(m) \\ &When\ \ N_{11}(m)/N(m) < N_{12}(m)/N(\neg m) \\ &\qquad E(m) = AIC\_DM(m) - AIC\_IM(m) \end{aligned} \tag{1}$$

Here, AIC_IM(s) and AIC_DM(s) are defined as follows [10].

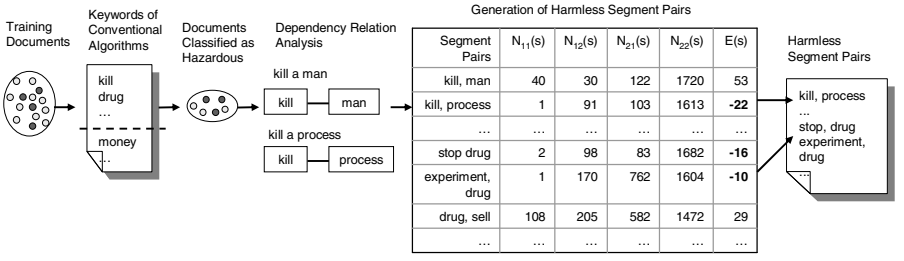
$$\begin{aligned} &AIC\_IM(m) = -2 \times MLL\_IM + 2 \times 2 \\ &MLL\_IM = N_p(m) \log N_p(m) + N(m) \log N(m) + N_n(m) \log N_n(m) \\ &\qquad + N(\neg m) \log N(\neg m) - 2N \log N \\ &AIC\_DM(m) = -2 \times MLL\_DM + 2 \times 3 \\ &MLL\_DM = N_{11}(m) \log N_{11}(m) + N_{12}(m) \log N_{12}(m) \\ &\qquad + N_{21}(m) \log N_{21}(m) + N_{22}(m) \log N_{22}(m) - N \log N \end{aligned} \tag{2}$$

Table 2 shows examples of keywords (morphemes) with high E(m) scores. We used 10,000 manually labeled Web pages[1] as training datasets (5,000 hazardous and harmless Web pages each). Here, Web pages that contain information on dating, criminal declarations, libelous statements and porn are labeled as hazardous. Table 2 shows how some keywords with high scores seem to be harmless. In contrast, our proposed algorithms aim to detect hazardous documents with high accuracy by using the keywords' neighboring segments.

3.3 Generation of Segment Pairs

Here we describe the algorithms for generating hazardous/harmless segment pairs from the conventional harmless/hazardous keyword sets. Figure 2 shows an overview

[1] Web pages are received from NetSTAR Inc. (http://www.netstar-inc.com/eng/) who engage in collection and manual classification of URLs.



Segment Pairs	N_{11}(s)	N_{12}(s)	N_{21}(s)	N_{22}(s)	E(s)
kill, man	40	30	122	1720	53
kill, process	1	91	103	1613	**-22**
…	…	…	…	…	…
stop drug	2	98	83	1682	**-16**
experiment, drug	1	170	762	1604	**-10**
drug, sell	108	205	582	1472	29
…	…	…	…	…	…

Fig. 2. Generation Algorithms of Harmless Segment Pairs

Table 3. Example of the Obtained Segment Pairs

Segment Pairs	N_{11}(s)	N_{12}(s)	N_{21}(s)	N_{22}(s)	E(s)
Actress, shot	106	144651	2	72293	74.7
Produce, Actress	0	144757	2	72293	-2.31
Sponsor, matchmaking	14	144743	1	72294	20.3
Sponsor, advertisement	2561	105965	708	10781	1129
Access, disguise	7	144750	3	72292	16.1
Access, guide	0	144757	27	72268	-9.20

of the generation algorithms. First, training datasets are classified as hazardous or harmless by the conventional algorithms. Dependency relations with hazardous keywords in sentences from hazardous documents are analyzed and segment pairs that contain hazardous keywords are extracted. The number of hazardous/harmless documents where each extracted segment pair "s" does/doesn't appear is evaluated in the same manner as in Table 1. In this case, the total number of documents N in Table 1 is defined as the number of hazardous documents. Harmless segment pairs are obtained by calculating an E(s) value based on expressions (1) and (2) in Section 3.1. In the same way, hazardous segment pairs are obtained from the harmless keywords with scores below the threshold, and training datasets are classified as harmless by the conventional algorithms. For example, in Figure 2, "kill" and "drug" are hazardous keywords, however, "kill a process" and "stop drug" are obtained as harmless segment pairs. When "money" has a below-threshold hazard score, the segment pair of "money" and "date" is hazardous. Table 3 shows examples of segment pairs obtained from 10,000 training data sets. Segment pairs with negative scores are harmless.

Here, we describe the appropriateness of using dependency relations to reflect contexts when detecting hazardous documents. The co-occurrence of morphemes is a possible approach to reflecting contexts. For example, "part-time girlfriend for a man" might be a hazardous sentence found in dating Web pages and the co-occurrence of "part-time", "girlfriend" and "man" are learned as hazardous. However, "a man's girlfriend quit her part-time" is a harmless sentence, yet it is regarded as harmful since it contains "part-time", "girlfriend" and "man". The dependency relationship between "part-time" and "girlfriend" does not appear in the latter sentence. Another approach to reflect contexts is using a simple n-gram; n adjacent morphemes. As a preliminary

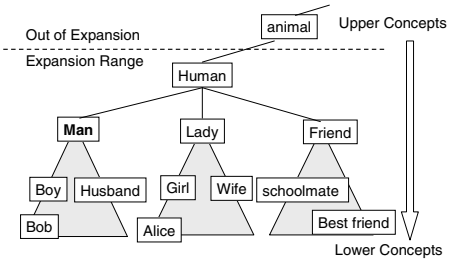


Fig. 3. Overview of the Base-line Expansion Algorithms of Segment Pairs

Fig. 4. Types and Appearance of Morphemes that Appear in Datasets

experiment, we evaluate the number of cases where 20 hazardous segment pairs appear adjacently in 10,000 Web pages (for example, in the sentence "part-time girlfriend for a man", "part-time" and "girlfriend" are adjacent). The segment pairs appear 311 times, and 202 times (64.9%) they are adjacent and 109 times (35.1%) they are apart. Our algorithms are particularly effective in cases of separated segments because a bi-gram approach cannot detect them.

3.4 Expansion with a Thesaurus

In order to adapt segment pairs extracted in Section 3.3 to more expressions, we expand the segment pairs with a thesaurus. In our algorithms, morphemes that are not listed in hazardous keyword sets are expanded. Figure 3 shows an overview of the base-line expansion algorithms in which morphemes are expanded to their one-level-higher concepts and their whole family of lower level concepts. For example, assume that "kill" is regarded as a harmless keyword in the conventional algorithms and a segment pair "kill" and "man" is extracted as hazardous by the proposed algorithms, then "man" is expanded to the higher concept "human" and its entire family of lower level concepts such as "lady", "Alice", and "Bob". Here we believe that the hazard score for "kill" is almost the same as for "man" for all lower level concepts of "human".

In our implementation of the proposed algorithms, we used the Japanese EDR thesaurus [12], which consists of 410,000 concepts and 270,000 words in a tree topology. Each entry contains a concept ID, concept title, concept explanation, ID list of the upper/lower concepts, and more. Each entry can contain several words. For example, the concept "school" has one higher concept, which is "building for education" and 14 lower concepts, including "elementary school", "university", and so forth. The concept "school" contains the word "school" only. Due to the large number of concept entries, making use of all entries is impractical due to the huge requirement for processing time and memory for the number of expanded segment pairs. In addition to the removal of ineffective expansion paths described previously, concept entries are removed whose words do not appear in the training datasets. In our preliminary experiment with 220,000 training datasets and 10,000 evaluation datasets, we evaluate the number of words both by type and appearance that appear

Table 4. Example of the Expanded Segment Pairs in Each Path

Expansion Path	Expansion Types	Path Contain Higher Layer	# of Down Layers	Correct	Incorrect	Correct Ratio
(1)	Harmless to Hazardous	No	1	433	82	84.1%
(2)		No	2	146	36	80.2%
(3)		No	3	58	25	69.9%
(4)		No	4	41	6	87.2%
(5)		No	5	0	0	-
(6)		Yes	1	27983	1254	95.7%
(7)		Yes	2	1852	349	84.1%
(8)		Yes	3	802	216	78.8%
(9)		Yes	4	551	143	79.4%
(10)		Yes	5	280	68	80.5%
(11)	Hazardous to Harmless	No	1	107	23	82.3%
(12)		...	...	...	...	...

solely in training datasets or evaluation datasets or both. Figure 4 shows that only a few words appear solely in evaluation datasets, which means there is almost no need for a concept entry whose words do not appear in training datasets. The number of words in the thesaurus is reduced from 270,000 to 25,000. We compare their performance by the required processing time and memory consumption in Section 4.

In addition, we focus on the expansion paths and their correctness. For example, a much lower concept may differ from the original concept. We classified expansion paths into 10 types depending on (a) whether they trace a one-level-higher concept, and (b) the depth of the paths they trace (up to 5). We defined expansion paths are correct when the expanded hazardous/harmless segment pairs appear in hazardous/harmless documents of the training datasets. Table 4 shows the number of expanded correct/incorrect segment pairs appearing in 200,000 Web pages. "Harmless to Hazardous" and "Hazardous to Harmless" mean that a document's classification as harmless/hazardous by the conventional algorithms is reversed by the proposed algorithms. In the "Harmless to Hazardous" case, paths (1), (2), (6), and (7) have high correctness levels which show expansion paths to the same or nearby levels have high correctness and paths to low levels have low correctness. "Hazardous to Harmless" has a similar tendency. In the experiment in Section 4, we confirm the improvement in the performance of the partial expansion algorithms that optimize the expansion paths by removing noisy paths, compared to the baseline algorithms of expanding to their one-level-higher concepts and all lower level concepts.

4 Performance Evaluation

4.1 Experimental Environments

Here we compare the performance of the conventional algorithms, the proposed base-line algorithms (BLA), the proposed base-line expansion algorithms (BLEA), and the proposed partial expansion algorithms (PEA). Experimental adjuncts and scenario are as follows.

Experimental Adjuncts
Japanese morphological analyzer: MeCab[13] Version 0.98, dictionary of morphological analyzer: IPADIC Version 2.7.0, (MeCab default), Japanese dependency analyzer: CaboCha[14] Version 0.53, dependency analysis models: CaboCha default, thesaurus: EDR thesaurus [12].

Datasets
240,000 manually labeled Web pages (220,000 training data; 110,000 hazardous and harmless Web pages each, 20,000 evaluation data; 10,000 hazardous and harmless Web pages each).

Criteria for Evaluation
We evaluate the recall rate, the precision rate, and the F value of each proposed algorithm and the conventional algorithms. In this paper, we define the recall, the precision, and F of the detection of hazardous Web pages as follows based on the total number of hazardous Web pages *All* (10,000 in this experiment), the number of Web pages detected as hazardous *Judge*, and the number of detected Web pages that are actually hazardous *Correct*.

$$Recall = Correct / All \tag{3}$$

$$Precision = Correct / Judge \tag{4}$$

$$F = 2 / (1 / Recall + 1 / Precision) \tag{5}$$

Experimental Scenario

1. Evaluate the trade-off of the recall rate and the precision rate of the conventional algorithms using several hazard score thresholds.
2. In each threshold of 1., correct the classification of the conventional algorithms by the proposed algorithms and evaluate the recall rate and the precision rate.
3. Expand the segment pairs with the thesaurus and evaluate the recall rate and the precision. We also evaluate the number of expanded segment pairs, the processing time and the memory consumption.

4.2 Experimental Results

Recall and precision rates of the conventional algorithms are shown in Figure 4. In the conventional algorithms, hazardous keywords are sorted by their hazard scores. With a high threshold, only a few hazardous keywords are used and the recall rate is low and the precision rate is high. Conversely, with a low threshold, the number of keywords used increases and the recall rate increases, but the precision decreases. We tested the proposed algorithms with 8 thresholds, producing the recall rates, the precision rates and the F values shown in Table 5.

Figure 5 shows the recall rate, the precision rate, and F value of the conventional algorithms, the BLA, and the BLEA. The improvement in the recall results from correcting the ‘harmless’ classifications by the conventional algorithms to ‘hazardous’. The improvement in precision results from correcting both ‘hazardous’ and ‘harmless’ classifications to ‘harmless’ and ‘hazardous’ respectively. The improvement of the BLA is up to 7.6% in the recall rate, up to 2.0% in the precision rate, and up to 4.8% in the F value. The improvement of the BLEA is up to 10.6% in

Table 5. Thresholds of the Conventional Algorithms and their Performance (%)

Threshold	# of Keywords	Recall	Precision	F
A	2	45.3	91.0	60.5
B	7	54.7	86.3	67.0
C	12	66.1	80.6	72.7
D	21	71.3	78.2	74.6
E	36	77.3	71.6	74.4
F	84	82.1	67.8	74.3
G	161	90.5	60.6	72.6
H	359	95.5	57.6	71.8

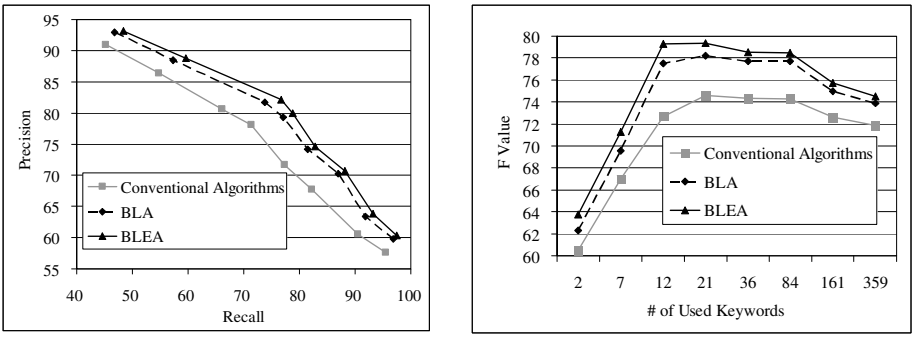


Fig. 5. Performance Comparison of the Conventional Algorithms, BLA, and BLEA. (Right. Recall vs. Precision, Left. F Value, %).

the recall rate, up to 3.2% in the precision rate, and up to 6.6% in the F value. This result means that segment pairs expanded by our algorithms detect more hazardous/harmless expressions.

In our experiments, we used the dependency analyzer CaboCha with default settings. The accuracy of dependency analysis of CaboCha for Web documents is reported as about 85% [15]. In our algorithms, dependency analysis errors reduced the number of extracted segment pairs in the training phase and reduced the number of corrections in the evaluation phase. In our experiments, however, the effectiveness of our algorithms was confirmed even with the default setting of the dependency analyzer. Improvement in the accuracy of dependency analysis for Web documents is expected to improve the performance of our algorithms.

Finally we evaluated the effect of removing unnecessary concepts and expanded noisy segment pairs based on their expanded path. Table 6 shows the number of segment pairs before/after expansion, processing time, and memory consumption. By removing unnecessary concepts, the average number of segment pairs is reduced to less than a quarter. The average processing time and memory consumption is also reduced about a quarter. Table 7 compares the performance of the BLEA and PEA. By removing several noisy expansion paths from Hazardous to Harmless segment pairs, the recall rate is improved by up to 1.70 and the F value improved by 0.75% compared to the peak performance of the BLEA which is 7.3% higher than the performance of the same number of keywords in the conventional algorithms.

Table 6. Number of Segment Pairs vs. Processing Time, and Memory Consumption

	BLA	BLEA			PEA		
Key-words	Seg. Pair (thousand)	Seg. Pair (million)	Time (sec)	Memory (GByte)	Seg. Pairs (million)	Time (sec)	Memory (GByte)
2	304	19.3	208	26.3	5.31	87	7.22
7	224	14	158	19	3.98	68	5.41
12	138	74.2	943	101	17	168	23.1
21	82.1	46	592	62.6	10.9	113	14.8
36	52	30.5	341	41.5	7.44	91	10.1
84	36.4	20.8	201	28.3	5.09	75	6.92
161	18.7	10.2	113	13.9	2.42	57	3.29
359	8.48	3.86	71	5.25	0.953	49	1.3
Avg.	108	27.4	328	37.2	6.63	88.5	9.02

Table 7. Performance Comparison of the BLEA vs. PEA (%)

	BLEA			PEA					
Key-words	Recall	Precision	F	Recall	Precision	F	Diff Recall	Diff Precision	Diff F
2	48.4	93.1	63.7	50.1	93.4	65.2	1.70	0.26	1.51
7	59.6	88.8	71.3	60.9	88.9	72.3	1.36	0.17	1.02
12	76.7	82.1	79.3	77.9	82.3	80.1	1.22	0.22	0.75
21	78.7	80.0	79.3	79.6	80.1	79.9	0.82	0.17	0.50
36	82.9	74.6	78.5	83.4	74.6	78.8	0.53	0.03	0.26
84	88.2	70.6	78.5	88.5	70.7	78.6	0.33	0.02	0.14
161	93.1	63.8	75.7	93.3	63.8	75.8	0.14	0.03	0.07
359	97.5	60.3	74.5	97.5	60.3	74.5	0	0	0

5 Conclusion

In this paper, we propose algorithms to increase the accuracy of hazardous Web page detection by correcting the classification of the conventional text-based algorithms based on the dependency relations of the hazardous keywords and their neighboring segments. In addition, we propose practical algorithms to increase performance by expanding the hazardous segment pairs using a thesaurus.

In our experiments with large scale real Web pages, the performance of the proposed base-line algorithms improved the performance of conventional algorithms by up to 6.6% in F value. Removing noisy segment pairs based on their expanded path is also effective which increased the peak performance of the base-line algorithms by 0.75 % and the improvement from the conventional algorithms was 7.3% in F value.

Acknowledgements. This research is supported by the National Institute of Information and Communications Technology, Japan.

References

1. Yanagihara, T., Ikeda, K., Matsumoto, K., Takishima, Y.: Fast n-gram Assortment Construction for Filtering Hazardous Information. IPSJ SIG Technical Reports, vol. 3, pp. 1–5 (2009)

2. Hoashi, K., Matsumoto, K., Inoue, N., Hashimoto, K.: Document Filtering Method Using Non-Relevant Information Profile. In: Proc. of SIGIR 2000, pp. 176–183 (2000)
3. Matsumura, A., Takasu, A., Adachi, J.: The Effect of Information Retrieval Method Using Dependency Relationship Between Words. In: Proc. of RIAO 2000, pp. 1043–1058 (2000)
4. Sun, R., Ong, C.H., Chua, T.S.: Mining Dependency Relations for Query Expansion in Passage Retrieval. In: Proc. of SIGIR 2006, pp. 382–389 (2006)
5. Liu, Y., Scheuermann, P., Li, X., Zhu, X.: Using WordNet to Disambiguate Word Senses for Text Classification. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2007. LNCS, vol. 4489, pp. 780–788. Springer, Heidelberg (2007)
6. Hsu, M.H., Tsai, M.F., Chen, H.H.: Combining WordNet and ConceptNet for Automatic Query Expansion: A Learning Approach. In: Li, H., Liu, T., Ma, W.-Y., Sakai, T., Wong, K.-F., Zhou, G. (eds.) AIRS 2008. LNCS, vol. 4993, pp. 213–224. Springer, Heidelberg (2008)
7. Yoshioka, M., Haraguchi, M.: On a Combination of Probabilistic and Boolean IR Models for WWW Document Retrieval. In: Proc. of TALIP 2005, vol. 4(4), pp. 340–356 (2005)
8. Li, S.L., Otsuka, M., Kitsuregawa, M.: Finding Related Search Engine Queries by Web Community Based Query Enrichment. In: Proc. of WWW 2010, pp. 121–142 (2010)
9. Ikeda, K., Yanagihara, T., Matsumoto, K., Takisima, Y.: Detection of Illegal and Hazardous Information Using Dependency Relations and Keyword Abstraction (in Japanese). In: Proc. of the Second Forum on Data Engineering and Information Management, C9-5 (2010)
10. Akaike, H.: A New Look at the Statistical Model Identification. IEEE Transactions on Automatic Control 19(6), 716–723 (2003)
11. Matsumoto, K., Hashimoto, K.: Schema Design for Causal Law Mining from Incomplete Database. In: Arikawa, S., Furukawa, K. (eds.) DS 1999. LNCS (LNAI), vol. 1721, pp. 92–102. Springer, Heidelberg (1999)
12. National Institute of Information and Communications Technology, "EDR Thesaurus", http://www2.nict.go.jp/r/r312/EDR/index.html
13. Kudo, T., Yamamoto, K., Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis. In: Proc. of EMNLP 2004, pp. 230–237 (2004), http://mecab.sourceforge.net/
14. Kudo, T., Yamamoto, K., Matsumoto, Y.: Japanese Dependency Analysis using Cascaded Chunking. In: Proc. of COLING 2002, pp. 63–69 (2002)
15. Kawahara, D., Kurohashi, S.: A fully-lexicalized probabilistic model for Japanese syntactic and case structure analysis. In: Proc. of NAACL 2010, pp. 176–183 (2010)

Improving Sentence Similarity Measurement by Incorporating Sentential Word Importance

Andrew Skabar and Khaled Abdalgader

Department of Computer Science and Computer Engineering
La Trobe University, Bundoora, Australia
a.skabar@latrobe.edu.au, kabdalgader@students.latrobe.edu.au

Abstract. Measuring similarity between sentences plays an important role in textual applications such as document summarization and question answering. While various sentence similarity measures have recently been proposed, these measures typically only take into account word importance by virtue of inverse document frequency (IDF) weighting. IDF values are based on global information compiled over a large corpus of documents, and we hypothesise that at the sentence level better performance can be achieved by using a measure of the importance of a word within the sentence that it appears. In this paper we show how the PageRank graph-centrality algorithm can be used to assign a numerical measure of importance to each word in a sentence, and how these values can be incorporated within various sentence similarity measures. Results from applying the measures to a difficult sentence clustering task demonstrates that incorporation of sentential word importance leads to statistically significant improvement in clustering performance as evaluated using a range of external clustering criteria.

1 Introduction

Measuring the similarity between sentences is an important function in applications such as extractive text summarization, where the objective is to extract from a document or documents a subset of sentences that are representative of the semantic content of the document(s) [1][2]. Although measuring text similarity at the document level is well-established in the Information Retrieval (IR) literature, where documents are represented in a common vector space [3], and similarity between them calculated using measures such as the Jaccard, Dice or Cosine measures [4], measuring similarity at the sentence level is complicated by the fact that two sentences may be semantically related despite having few, if any, words in common. Consequently, various measures for short text similarity have been recently proposed [5][6][7][8]. To overcome the problem of lack of word co-occurrence, these methods are typically based on word-to-word similarity measures derived either from distributional information from some corpora (corpus-based measures), or semantic information represented in external sources such as WordNet [9] (knowledge-based measures).

While it is widely accepted that incorporating inverse document frequency (IDF) scores leads to improved measurement of text similarity at the document level, it is not clear that it has the same utility at the sentence level. For example, in evaluating

J. Li (Ed.): AI 2010, LNAI 6464, pp. 466–475, 2010.

the performance of a variety of sentence similarity measures on a range of tasks, Achananuparp *et al*. (2008) [10] report that measures such as IDF have no clear advantage in the overall performance of these similarity measures.

Other difficulties are also inherent in using IDF scores at the sentence level. For example, many words are polysemous (i.e., have multiple meanings), and we may wish to disambiguate words prior to measuring similarity between them. The problem is that IDF scores are generally not available for specific senses of words, and therefore use of non sense-assigned IDF values only serves to add noise.

In this paper, we explore the idea of incorporating into sentence similarity measures a factor based on the importance of words in the sentences being compared. We refer to this importance as 'sentential word importance' to distinguish it from measures such as IDF, which are derived from large corpora. Specifically, we show how the PageRank algorithm [11] can be used to assign a numerical measure of importance to each word in a sentence. We then show how these importance measures can be incorporated into several existing sentence similarity measures. We apply the techniques to a difficult sentence clustering task, and show that incorporating sentential word importance leads to statistically significant improvement in clustering performance, as evaluated using a range of clustering criteria.

The remainder of the paper is structured as follows. Section 2 describes several existing sentence similarity measures. Section 3 describes how PageRank can be used to determine sentential word importance, and how this word importance can be incorporated into existing similarity measures. Section 4 provides empirical results and Section 5 concludes the paper.

2 Sentence Similarity Measures

A variety of sentence similarity measures have been proposed in recent years. In this section we describe two common measures, and propose a third. In Section 3 we will describe how these measures can be improved through incorporation of sentential word importance.

The sentence similarity measure proposed by Mihalcea *et al.* (2006) [6] operates as follows. Given two sentences S_1 and S_2, first calculate the similarity between the first word in S_1 and each word in S_2 that belongs to the same part of speech class. The maximum of these scores is then weighted with the *IDF* score of the word from S_1. This procedure is then repeated for the remaining words in S_1, with the weighted maximum scores summed, and then normalized by dividing by the sum of *IDF* scores. This entire procedure is then repeated for S_2. The overall similarity is finally defined as the average of normalized weighted maximums for S_1 and S_2. In mathematical notation:

$$sim(S_1, S_2) = \frac{1}{2} \sum_{w \in \{S_1\}} \left(\arg\max_{x \in \{S_2\}} sim(w, x) \times idf(w) \right) \Bigg/ \sum_{w \in \{S_1\}} idf(w) + \frac{1}{2} \sum_{w \in \{S_2\}} \left(\arg\max_{x \in \{S_1\}} sim(w, x) \times idf(w) \right) \Bigg/ \sum_{w \in \{S_2\}} idf(w) \quad (1)$$

where $sim(x, y)$ is the similarity between words x and y. The *IDF* score is determined using an external corpus. The reason for computing the semantic similarity scores only between

words in the same part of speech class is that most WordNet-based measures are unable to calculate semantic similarity of words belonging to different parts of speech.

Whereas the above approach does not utilize any explicit sentence representation, the approach proposed by Li *et al*. (2006) [5] is similar to that used to calculate document similarity in the Information Retrieval (IR) literature. However, rather than using a common vector space representation for all sentences, the two sentences being compared are represented in a reduced vector space of dimension n, where n is the number of distinct words in the union of the two sentences. Semantic vectors, $\mathbf{V}_1$ and $\mathbf{V}_2$, are first constructed. These vectors represent sentences S_1 and S_2 in the reduced space. The similarity between S_1 and S_2 is then defined as the Cosine similarity [4] between $\mathbf{V}_1$ and $\mathbf{V}_2$. The elements of $\mathbf{V}_i$ are determined as follows. Let v_{ij} be the j^{th} element of $\mathbf{V}_i$, and let w_j be the word corresponding to dimension j in the reduced vector space. There are two cases to consider, depending on whether w_j appears in S_i:

Case 1: If w_j appears in S_i, set v_{ij} equal to 1.

Case 2: If w_j does not appear in S_i, calculate a word-word semantic similarity score between w_j and each non-stopword in S_i, and set v_{ij} to the highest of these similarity scores i.e., $v_{ij} = \arg\max_{x \in \{S_i\}} sim(w_j, x)$.

We note that in their formulation Li *et al*. (2006) [5] also factor in an *information content* weighting so that the similarity between two words i and j is defined as $sim(w_i, w_j) \times I(w_i) \times I(w_j)$, where $sim(w_i, w_j)$ is defined as above, and $I(w)$ is the information content of word w, and is defined as $-\log p(w) / \log(N+1)$ where $p(w)$ is the probability that the word appears in a large corpus and N is the total number of words in the corpus. As a measure of word importance, information content, therefore, plays a similar role to IDF. Li *et al*. (2006) [5] also incorporate a factor based on word order.

The third measure we investigate is derived from distance measures used in the clustering literature, and we are not aware of it having been used in the literature on sentence similarity measures. The measure, which we refer to as the *group average* method, is straightforward, and simply sums the similarities between all words in the first sentence with all words in the second, normalizing by the size of the sentences:

$$sim(S_1, S_2) = \frac{\sum_{w_1 \in \{S_1\}} \sum_{w_2 \in \{S_2\}} sim(w_1, w_2)}{|S_1| \times |S_2|} \tag{2}$$

Unlike the first two measures described above, which are based on the *maximum* similarity between a word and the words of the opposing sentence, the group average method uses all similarities. Intuitively, because it uses all similarity information, one might expect it to perform at least as well as the other measures.

2.1 Word-to-Word Semantic Similarity Measures

Each of the above measures relies on a measure of word-to-word similarity. Various word-to-word measures have been proposed, and these can broadly be categorized as being either *corpus-based* or *knowledge-based*. Whereas corpus-based measures are based on distributional information derived from large corpora such as the Brown Corpus, knowledge-based measures utilize semantic relations expressed in resources such as dictionaries or thesauri. We use the latter approach. Specifically, we use

lexical knowledge-base WordNet [9] in conjunction with the word-to-word semantic similarity measure due to Jiang and Conrath [12].

The Jiang and Conrath measure is based on the idea that the degree to which two words are similar is proportional to the amount of information they share. The similarity between words w_1 and w_2 is defined as:

$$Sim_{J\&C}(w_1, w_2) = \frac{1}{IC(w_1) + IC(w_2) - 2 \times IC(LCS(w_1, w_2))} \tag{3}$$

where $LCS(w_1, w_2)$ is the word that is the *deepest common ancestor* of w_1 and w_2 in the WordNet hierarchy, $IC(w)$ is the *information content* of word w, and is defined as $IC(w) = -\log P(w)$, where $P(w)$ is the probability that word w appears in a large corpus (e.g., the Brown corpus). For a comprehensive review of word similarity measures, see Budanitsky & Hirst (2006) [13].

3 Graph-Based Word Importance Ranking

Algorithms such as PageRank [11] and HITS [14] belong to the family of graph-based ranking algorithms. The basic idea behind these algorithms is that the importance of a vertex within a graph can be determined by taking into account global information recursively computed from the entire graph. It is this importance that can be used a measure of centrality. We focus here on PageRank.

PageRank assigns to every node in a graph a numerical score between 0 and 1, known as its *PageRank score* (*PR*). This score is defined as

$$PR(V_i) = (1 - d) + d \times \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} PR(V_j) \tag{4}$$

where $In(V_i)$ is the set of vertices that point to V_i, $Out(V_j)$ is the set of vertices pointed to by V_j, and d is a damping factor, typically set to around 0.8 to 0.9 [11]. Using the analogy of a random surfer on the Web, nodes visited more often will be those with many links coming in from other frequently visited nodes, and the role of d is to reserve some probability for jumping to any node in the graph, thereby preventing the user getting stuck in some disconnected part of the graph.

Although proposed in the context of Web page ranking, PageRank can be used more generally to determine the importance (or *centrality*) of an object in a network. For example Mihalcea and Tarau's (2004) TextRank [15] and Erkan and Radev's (2004) LexRank [2] both use PageRank for ranking sentences for the purpose of extractive text summarization. In these systems, each sentence in a document or documents is represented by a node on a graph. However, unlike a Web graph, in which edges are directed but unweighted, edges on a document graph are weighted with a value representing the similarity between sentences. These edges are usually undirected, since similarity is usually a symmetric relation; however it is also possible to use directed edges. The modification of PageRank to deal with weighted, undirected edges is straightforward:

$$PR(V_i) = (1-d) + d \times \sum_{j=1}^{N} \left(w_{ji} \frac{PR(V_j)}{\sum_{k=1}^{N} w_{jk}} \right) \tag{5}$$

where w_{ij} is the similarity between V_i and V_j. Note that the summations are now over all vertices in the graph. If weights are binary, then the formula reduces to the original.

It is straightforward to extend this idea to representing a sentence as a graph in which vertices are words, and edge weights represent the similarity between words. Edge weights can be determined using word-to-word similarity measures such as the Jiang and Conrath measure. Equation 5 can then be used to assign to each word a score representing the importance of that word in the sentence.

As an example, consider the following two sentences, both taken from the famous quotations dataset which we have compiled (see Section 4):

S1: "A deaf husband and a blind wife are always a happy couple."

S2: "The woman cries before the wedding; the man afterward."

After performing word sense identification using a variant of the Lesk method [16], calculating pairwise similarities between the resulting sense-assigned words using the method described in Section 2.1, and finally applying PageRank, we obtain the following, where numbers below the word/sense pairs are the corresponding PageRank values.

S1: [('deaf', 0), ('husband', 0), ('blind', 0), ('wife', 0), ('happy', 1), ('couple', 1)]
[0.1106 0.2479 0.1157 0.2506 0.16667 0.1085]

S2: [('woman', 0), ('cries', 4), ('wedding', 2), ('man', 0), ('afterward', 0)]
[0.2866 0.1170 0.1096 0.2868 0.2000]

The words found to be most central in Sentence 1 are 'husband' and 'wife', and the words most central in Sentence 2 are 'woman' and 'man'. Both of these sentences are about marriage, and the concepts of man/woman and husband/wife are clearly related to the concept of marriage. We hypothesize that incorporating these PageRank scores into the measurement of sentence similarity will result in an improved measure.

3.1 Modified Sentence Similarity Measures

Incorporating PageRank values into the measures described in Section 2 is relatively straightforward. The measure proposed by Mihalcea *et al.* (2006) [6] can be modified as follows:

$$sim(S_1, S_2) = \frac{1}{2} \sum_{w \in \{S_1\}} \left(sim\left(w, \arg\max_{x \in \{S_2\}} \left(sim(w,x) \times PR_x^{S_2} \right) \right) \times PR_w^{S_1} \right) \Big/ \sum_{w \in \{S_1\}} PR_w^{S_1} \; + \\ \frac{1}{2} \sum_{w \in \{S_2\}} \left(sim\left(w, \arg\max_{x \in \{S_1\}} \left(sim(w,x) \times PR_x^{S_1} \right) \right) \times PR_w^{S_2} \right) \Big/ \sum_{w \in \{S_2\}} PR_w^{S_2} \tag{6}$$

where PR_x^S is the PageRank score of word x in sentence S. Note that this incorporates the PageRank of both the target word (i.e., words appearing in the outer summations), as well as the PageRank values of the words against which the target words are being compared.

For the Li *et al*. (2006) [5] measure, the only modification required is in determining the components of the semantic vectors. This is done as follows:

Case 1: If w_j appears in S_i, set v_{ij} equal to $PR_{w_j}^{S_i}$ (i.e., the PageRank score for w_j in S_i).

Case 2: If w_j does not appear in S_i, set v_{ij} equal to the highest weighted similarity between w_j and the words in S_i; i.e., $v_{ij} = \arg\max_{x \in \{S_i\}} \left(sim(w_j, x) \times PR_{w_j}^{S_i} \right)$.

For the third approach, we weight the word-to-word similarity measures by the PageRank values of the words in their respective sentences, and normalize by the product of summed PageRank values for each sentence.

$$sim(S_1, S_2) = \frac{\sum_{w_1 \in \{S_1\}} \left(\sum_{w_2 \in \{S_2\}} \left(sim(w_1, w_2) \times PR_{w_2}^{S_2} \right) \times PR_{w_1}^{S_1} \right)}{\sum_{w_2 \in \{S1\}} PR_{w_1}^{S_1} \times \sum_{w_2 \in \{S_2\}} PR_{w_2}^{S_2}} \tag{7}$$

4 Empirical Results

Sentence similarity measures are commonly evaluated on binary classification tasks such as the Microsoft Research Paraphrase Corpus (MSRP) [17], or the RTE datasets [18]. However, there are a number of problems with evaluating similarity measures in this way. Firstly, performing binary classification requires that a threshold be determined, and this requires a training set. Most researchers who have used these datasets are interested only in unsupervised learning, and usually choose a threshold of 0.5. This choice, however, is *ad hoc*, since similarity measures such as those we have described do not output probabilities; moreover, some measures output a consistently higher range of values than others. Secondly, performing binary classification does not test the full discriminatory capability of a similarity measure. For example, if a measure achieves good performance on a classification task, it does not necessarily follow that the measure will achieve good performance when used within some other task, such as clustering.

4.1 Famous Quotations Dataset

Since our interest in measuring sentence similarity is motivated by our work in sentence clustering, we choose to test the similarity measures on the task of clustering famous quotations. Quotations provide a rich and challenging context for the evaluation of sentence clustering performance because they often contain a lot of semantic information (i.e., wisdom packed into a small message), and are often couched in a poetic use of language. We have compiled a dataset consisting of 50 famous quotes from 5 different classes (marriage, peace, food, nature, knowledge) (10 from each class). The quotes are taken from the website in [19]. An excerpt is shown in Table 1.

4.2 Spectral Clustering

Since the sentence similarity measures do not represent sentences in a common and continuous feature space, popular clustering methods such as *K*-means, Fuzzy *K*-means

and mixture models, all of which assume that points lie in a metric space, and take rectangular attribute data as input, are no longer generally applicable. Instead, we require a clustering algorithm which takes as input only pairwise similarities between objects (which may be represented in a similarity matrix). Spectral Clustering algorithms, which belong to the family of graph-based approaches and which have become one of the most popular clustering methods in recent years, satisfy the criterion of only requiring pairwise similarities as input. These algorithms are based on matrix decomposition techniques: rather than clustering data points in the original vector space, data points are mapped onto the space defined by the eigen-vectors associated with the top eigen-values, and clustering is then performed in this transformed space, typically using a *K*-means algorithm. A variety of spectral clustering algorithms have been proposed, and in this paper we use the algorithm due to Ng *et al*. (2001) [20]. Detailed information on spectral clustering can be found in Luxburg (2007) [21].

Table 1. Extract from Famous Quotations Dataset

Knowledge
1. Our knowledge can only be finite, while our ignorance must necessarily be infinite.
2. Pocket all your knowledge with your watch and never pull it out in company unless desired.
3. Knowledge is of two kinds; we know a subject ourselves, or we know where we can find information upon it.
…
Marriage
10. A husband is what is left of a lover, after the nerve has been extracted.
11. A rich widow weeps with one eye and signals with the other.
12. A wise woman will always let her husband have her way.
…
Nature
21. The course of nature is the art of God.
22. From the intrinsic evidence of His creation, the Great Architect of the Universe now begins to appear as a pure mathematician.
23. Nature, with equal mind, sees all her sons at play, sees man control the wind, the wind sweep man away.
…
Peace
31. There is no such thing as inner peace, there is only nervousness and death.
32. When fire and water are at war it is the fire that loses.
33. Once you hear the details of victory, it is hard to distinguish it from a defeat.
…
Food
41. Food is an important part of a balanced diet.
42. At the end of every diet, the path curves back toward the trough.
43. Hunger is not debatable.
…

4.3 Clustering Evaluation Criteria

Two widely used external cluster quality measures are *purity* and *entropy*. The purity of a cluster is the fraction of the cluster size that the largest class of objects assigned to that cluster represents. Overall purity is the weighted sum of the individual cluster purities and is given by

$$Purity = \frac{1}{N}\sum_{j=1}^{|L|}\left(|w_j| \times P_j\right) \text{ where } P_j = \frac{1}{|w_j|}\max_i\left(|w_j \cap c_i|\right).$$

$L = \{w_1, w_2, \ldots\}$ is the set of clusters, $C = \{c_1, c_2, \ldots\}$ is the set of classes, and N is the number of objects.

Entropy is defined as weighted average of the individual cluster entropy over all clusters:

$$Entropy = \sum_{j=1}^{|L|} \frac{|w_j|}{N} \left(-\frac{1}{\log|C|} \sum_{i=1}^{|C|} \frac{|w_j \cap c_i|}{|w_j|} \log \frac{|w_j \cap c_i|}{|w_j|} \right)$$

Because entropy and purity measure how the classes of objects are distributed within each cluster, they measure *homogeneity*; i.e., the extent to which clusters contain only objects from a single class. However, we are also interested in *completeness*; i.e., the extent to which all objects from a single class are assigned to a single cluster. While high purity and low entropy are generally easy to achieve when the number of clusters is large, this will result in low completeness, and in practice we are usually interested in achieving an acceptable balance between the two.

The *V*-measure, also known as the Normalized Mutual Information (NMI) [7], takes into account both homogeneity (h) and completeness (c), and is defined as the harmonic mean of the two: $V = hc / (h + c)$, where h and c are defined as

$$h = 1 - \frac{H(C|L)}{H(C)} \quad \text{and} \quad c = 1 - \frac{H(L|C)}{H(L)}$$

where

$$H(C) = -\sum_{i=1}^{|C|} \frac{|c_i|}{N} \log \frac{|c_i|}{N}, \qquad H(L) = -\sum_{j=1}^{|L|} \frac{|w_j|}{N} \log \frac{|w_j|}{N},$$

$$H(C|L) = -\sum_{j=1}^{|L|} \sum_{i=1}^{|C|} \frac{|w_j \cap c_i|}{N} \log \frac{|w_j \cap c_i|}{|w_j|}, \qquad H(L|C) = -\sum_{i=1}^{|C|} \sum_{j=1}^{|L|} \frac{|w_j \cap c_i|}{N} \log \frac{|w_j \cap c_i|}{|c_i|}.$$

Unlike the statistical approaches described above, Rand Index and *F*-measure are based on a combinatorial approach which considers each possible pair of objects. Each pair can fall into one of four groups: if both objects belong to the same class and same cluster, then the pair is a true positive (TP); if objects belong to the same cluster but different classes, the pair is a false positive (FP); if objects belong to the same class but different clusters, the pair is a false negative (FN); otherwise the objects belong to different classes and different clusters, in which case the pair is a true negative (TN). The Rand index is simply the accuracy; i.e., $(TP + FP)/(TP + FP + FN + TN)$. The *F*-measure is the harmonic mean of precision and recall; i.e., $2PR/(P+R)$, where $P = TP/(TP + FP)$ and $R = TP/(TP + FN)$.

4.4 Results

Table 2 shows the results of applying the original measures described in Section 2 (labelled as unweighted), and the modified measures proposed in Section 3 (weighted). In each case we performed 200 trials, each with a different random initialization for the clustering algorithm. Upper numbers are means; numbers in parentheses are standard deviations. On all three similarity measures, and for all clustering evaluation measures, the improvement in clustering performance achieved through weighting by sentential word importance was found to be statistically significant at the 0.05 level using a two-sample two-tailed t-test. (In most cases the results were significant at the 0.01 level or better). Since inclusion or exclusion of

word importance was the only variable, we conclude that incorporating sentential word importance does lead to improvement in sentence similarity measurement.

Best overall performance is achieved using the modified version of Li *et al.*'s [5] measure, and this is followed closely by the modified version of the Mihalcea *et al.* [6] measure. These measures have in common the fact that they are based on the maximum semantic similarity between a word and the words in the opposing sentence. In contrast, the group average measure averages over all inter-sentence word-to-word similarities, and intuitively one might expect it to be a superior measure on account of this. However its performance—both for the weighted an unweighted versions—is inferior to both the other methods. We speculate that spurious word-to-word similarities, sometimes originating from incorrect word sense identification, result in the addition of noise, thereby leading to its inferior performance.

Table 2. Clustering Performance on Famous Quotations Dataset

	Purity	**Entropy**	***V*-measure**	**Rand Index**	***F*-measure.**
Modified Li *et al.* (2006) Method					
Unweighted	0.650 (0.037)	0.508 (0.022)	0.509 (0.020)	0.783 (0.016)	0.476 (0.024)
Weighted	0.679 (0.058)	0.471 (0.056)	0.542 (0.053)	0.805 (0.026)	0.516 (0.049)
Modified Mihalcea *et al.* (2006) Method					
Unweighted	0.648 (0.012)	0.515 (0.017)	0.521 (0.015)	0.738 (0.022)	0.443 (0.020)
Weighted	0.695 (0.046)	0.486 (0.058)	0.530 (0.047)	0.781 (0.048)	0.483 (0.065)
Group Average Method					
Unweighted	0.565 (0.009)	0.605 (0.010)	0.450 (0.012)	0.646 (0.013)	0.372 (0.007)
Weighted	0.575 (0.026)	0.544 (0.013)	0.506 (0.016)	0.694 (0.008)	0.411 (0.008)

In regard to increased computational cost, we note that the step of calculating sentential word importance contributes extremely little. Graph-based centrality measures such as PageRank converge quickly, even for a relatively large number of nodes. Assuming that sentences contain in the order of 10 or so words, any increase in computational cost is virtually insignificant when compared, for example, against the cost of computationally expensive tasks such as word sense disambiguation.

5 Conclusion

The idea of incorporating word importance in text similarity measurement is not new, and IR researchers have been using IDF weights in measuring document similarity for decades. However, IDF weights are determined using an external corpus, and while this may provide information on how important a word is when taken over a large corpus, it provides little information on the importance of a word in the context of the sentence in which it appears. We have described how the PageRank algorithm can be used to determine sentential word importance, and how the resulting importance scores can then be incorporated into a variety of sentence similarity measures. Testing of the methods

on a difficult sentence clustering task has demonstrated that incorporating word importance leads to a significant improvement in clustering performance.

References

1. Nomoto, M.: A New Approach to Unsupervised Text Summarization. In: Proceedings of the 24th ACM SIGIR, pp. 26–34 (2001)
2. Erkan, G., Radev, D.: LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. Journal of Art. Int. Research 22, 457–479 (2004)
3. Salton, G.: Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley, Reading (1989)
4. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
5. Li, Y., McLean, D., Bandar, Z., O'Shea, F., Crockett, K.: Sentence Similarity Based on Semantic Nets and Corpus Statistics. IEEE TKDE 18(8), 1138–1150 (2006)
6. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and Knowledge-based Measures of Text Semantic Similarity. In: 21st National Conference on Art. Int., Boston, vol. 1, pp. 775–780 (2006)
7. Islam, A., Inkpen, D.: Semantic Text Similarity using Corpus-based Word Similarity and String Similarity. ACM Trans. on KDD 2(2), 1–25 (2008)
8. Achananuparp, P., Hu, X., Yang, C.: Addressing the Variability of Natural Language Expression in Sentence Similarity with Semantic Structure of the Sentences. In: PAKDD, pp. 548–555 (2009)
9. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge
10. Achananuparp, P., Hu, X., Shen, X.: The Evaluation of Sentence Similarity Measures. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2008. LNCS, vol. 5182, pp. 305–316. Springer, Heidelberg (2008)
11. Brin, S., Page, L.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. Computer Networks and ISDN Systems 30(1-7), 107–117 (1998)
12. Jiang, J.J., Conrath, D.W.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: 10th Inter. Conf. on Research in Computational Linguistics, pp. 19–33 (1997)
13. Budanitsky, A., Hirst, G.: Evaluating WordNet-based Measures of Lexical Semantic Relatedness. Computational Linguistics 32(1), 13–47 (2006)
14. Kleinberg, J.M.: Authoritative Sources in a Hyperlinked Environment. Journal of the ACM (JACM) 46(5), 604–632 (1999)
15. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In: EMNLP, pp. 404–411 (2004)
16. Lesk, M.: Automatic Sense Disambiguation using Machine Readable Dictionaries: How to tell a pine cone from an ice cream cone. In: Proc. of the SIGDOC, pp. 24–26 (1986)
17. Dolan, W., Chris Quirk, C., Brockett, C.V.: Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. In: 20th International Conf. on Computational Linguistics, pp. 350–356 (2004)
18. Dagan, I., Dolan, B., Giampiccolo, D., Magnini, B.: The Third PASCAL Recognizing Textual Entailment Challenge. In: ACL-PASCAL Workshop on TEP, pp. 1–9 (2007)
19. `http://www.famousquotesandauthors.com/` (accessed May 26, 2010)
20. Ng, A.Y., Jordan, M.I., Weiss, Y.: On Spectral Clustering: Analysis and an Algorithm. In: NIPS, pp. 849–856 (2001)
21. Luxburg, V.: A Tutorial on Spectral Clustering. Statistics and Computing 17(4), 395–416 (2007)

Collaborative Filtering for People to People Recommendation in Social Networks

Xiongcai Cai, Michael Bain, Alfred Krzywicki, Wayne Wobcke,
Yang Sok Kim, Paul Compton, and Ashesh Mahidadia

School of Computer Science and Engineering,
University of New South Wales, Sydney NSW 2052, Australia
{xcai,mike,alfredk,wobcke,yskim,compton,ashesh}@cse.unsw.edu.au

Abstract. Predicting people other people may like has recently become an important task in many online social networks. Traditional collaborative filtering approaches are popular in recommender systems to effectively predict user preferences for items. However, in online social networks people have a dual role as both "users" and "items", e.g., both initiating and receiving contacts. Here the assumption of active users and passive items in traditional collaborative filtering is inapplicable. In this paper we propose a model that fully captures the bilateral role of user interactions within a social network and formulate collaborative filtering methods to enable people to people recommendation. In this model users can be similar to other users in two ways – either having similar "taste" for the users they contact, or having similar "attractiveness" for the users who contact them. We develop SocialCollab, a novel neighbour-based collaborative filtering algorithm to predict, for a given user, other users they may like to contact, based on user similarity in terms of both attractiveness and taste. In social networks this goes beyond traditional, merely taste-based, collaborative filtering for item selection. Evaluation of the proposed recommender system on datasets from a commercial online social network show improvements over traditional collaborative filtering.

Keywords: Machine Learning, Recommender Systems, Collaborative Filtering.

1 Introduction

Traditional social filtering or *recommender* systems attempt to discover user preferences over items by modelling the relation between users and items. The aim is to recommend items that match the *taste* (likes or dislikes) of users in order to assist the active user, i.e., the user who will receive recommendations, to select items from an overwhelming set of choices. Such systems have many uses in e-commerce, subscription based services and other online applications, where provision of personalised suggestions is required [8]. By applying recommendation techniques it is possible to greatly increase the likelihood of the successful purchase of products or services by the active user, since services or products are

J. Li (Ed.): AI 2010, LNAI 6464, pp. 476–485, 2010.

personalised and presented to the active user using information obtained from the purchasing behaviour of like-minded users. In online applications with a very large number of choices where customer taste is important in making selections, personalised recommendation of items or people becomes essential.

1.1 Recommender Systems

Approaches to recommender systems can be categorised as content-based or collaborative filtering methods. In content-based methods, the user will be recommended items similar to those the user preferred in the past. This is usually based on models created from item descriptions using information retrieval or machine learning techniques. In general, a content-based system analyses the content of the profiles, or descriptions, of items, as well as provided user ratings, to infer a model that can be used to recommend additional items of interest. In this paper we do not address content-based recommendation.

Collaborative filtering (CF) methods, on the other hand, recommend items based on aggregated user preferences of those items, which does not depend on the availability of item descriptions. In CF, preference information from a set of users is utilised to make automatic predictions about the interests of the active user by assuming that user preferences hold over time. Importantly, predictions are made by models personalised to the taste of each active user based on information from many users, rather than from a global model making predictions for all users.

Collaborative filtering algorithms fall into two categories: memory-based and model-based approaches. Memory-based approaches [1,4,5,7] use heuristics to make rating predictions based on the entire collection of items previously rated by users. The unknown rating value $r_{c,s}$ of the active user c for an item s is typically computed as an aggregate of the ratings of users similar to c for the same item s. This aggregate can be an average or a weighted sum, where the weight is a distance that measures the similarity $sim(c_1, c_2)$ between users c_1 and c_2.

In contrast, model-based CF approaches [1,2,3,6,9] use the collection of ratings to learn a model, which is then used to make rating predictions. Although model-based methods have reported better accuracy of recommendation than memory-based approaches, there are also some limitations. Firstly, these methods are computationally expensive since they usually require all users and items involved to be used in creating models. Secondly, they attempt to predict the *rating* of a user rather than correctly *rank* the items.

1.2 People to People Recommendation

In this paper, we propose a recommendation method for people to people recommendation in social networks. In the traditional scenario where CF is applied, only the taste of users counts and items are passive in terms of the business transaction, i.e., once a user selects an item there is no response by that item. However in social networks, "items" are also users who actively participate in

social interactions. In this sense, traditional CF is not applicable for people to people recommendation, since it only considers the taste of one side. We propose in this paper to extend traditional CF methods so the recommender system will handle the bilateral nature of such interactions.

We propose *SocialCollab*, a novel neighbour-based collaborative filtering algorithm to predict, for a given user, other users they may like to contact. This recommender system is based on the similarity of users in terms of the bilateral properties of attractiveness and taste. The main contribution of this paper is a novel approach for recommendation of potential friends or partners based on a new formalisation of the bilateral nature of interaction in social networks.

The paper is organised as follows. Section 2 presents a bilateral collaborative filtering framework for recommendation in social networks. Experimental evaluation is in Section 3 and conclusions are in Section 4.

2 Bilateral Collaborative Filtering

2.1 A Prototypical Collaborative Filtering Algorithm

Traditional collaborative filtering can operate in two directions: user-based or item-based. User-based approaches look for users who share the same rating patterns with the active user (the user whom the prediction is for) and then uses the ratings from like-minded users to calculate a prediction for the active user. On the other hand, item-based collaborative filtering such as that of Amazon.com [5] creates an item-item matrix determining relationships between pairs of items, which is then used to infer the taste of the active user.

The most important step in both approaches is determining similarity. Two items are *similar* if both are selected together by a set of users. Alternatively, two users are similar if they both select the same set of items (i.e., they have similar taste). The underlying assumption of CF approaches is that those who agreed in the past tend to agree again in the future. User-based approaches assume that two users will like the same items if they have similar taste. Therefore, an item is potentially recommended to the active user if it is selected by a similar user:

$$i \Rightarrow u : \exists s, (s \leftrightarrow u \wedge s \rightarrow i) \tag{1}$$

where $i \Rightarrow u$ denotes recommending i to u, $s \leftrightarrow u$ denotes that s is similar to u and $s \rightarrow i$ represents that s selected i.

Item-based approaches assume items can be related by the fact that they are frequently selected together by users, and will recommend an item which is similar to items that the active user selected:

$$i \Rightarrow u : \exists s, (s \leftrightarrow i \wedge u \rightarrow s) \tag{2}$$

These assumptions are only valid for recommending items to users where the selection is determined only by the user, not the item. In social networks, this is not the case – there is a two-way interaction. For user recommendation in social networks, collaborative filtering needs to be extended, as described in the next section.

2.2 Collaborative Filtering for Social Networks

In social networks, "items" as the recipients of actions are also *users* who are actively participating in social interactions. When they are contacted by other users, they can make different responses, either positive or negative. Therefore, traditional CF is not applicable to people recommendation since it only considers the taste of one side (users) and neglects the other (items). The recommender framework needs to be extended to handle the bilateral nature of such interactions in people recommendation.

Successful Interaction. We define an successful interaction as:

Definition 1. *An interaction between two users is a successful interaction when it has a positive response.*

Positive responses are usually defined in the application domain. For example, in an online dating site, a user Bob can send a message to another user Alice to express his interest in her. This message is a *contact*. This contact creates an *interaction* once it receives a corresponding *reply*. If the reply is positive, i.e. Alice also expresses her interest in Bob, this interaction becomes a successful interaction. Otherwise, it is an *unsuccessful interaction*.

User Attractiveness and Taste. In people recommendation, users have taste that determines their favourites when they actively make decisions selecting other users. At the same time, users are also passively involved in interactions by being selected by other users, which reflects, in some sense, their *attractiveness* within the social network. In this regard, both the aspects of users' taste and attractiveness need to be modelled. We define the similar attractiveness and similar taste of users as follows.

Definition 2. *Two users are similar in attractiveness ($u_i \overset{a}{\leftrightarrow} u_i$) if they are both selected by a nonempty set of users in common:*

$$u_i \overset{a}{\leftrightarrow} u_j : \exists U, (U \rightarrow u_i \wedge U \rightarrow u_j). \tag{3}$$

Definition 3. *Two users are similar in taste ($u_i \overset{t}{\leftrightarrow} u_j$) if they both select a nonempty set of users in common:*

$$u_i \overset{t}{\leftrightarrow} u_j : \exists U, (u_i \rightarrow U \wedge u_j \rightarrow U) \tag{4}$$

SocialCollab: Modelling Bilateral Decisions. Just because the active user likes a user does not mean a successful match since the liked user may not like the active user. This requires that the liked user also likes the active user. The point here is that only when u_a likes u_r and also u_r likes u_a can an interaction be a success. Only in this case, u_r should be recommended to u_a.

To model this behaviour, following traditional collaborative filtering assumptions, we define the following two assumptions based on user taste:

1. If people with similar taste to u_a like u_r, u_a will like u_r;
2. If people with similar taste to u_r like u_a, u_r will like u_a.

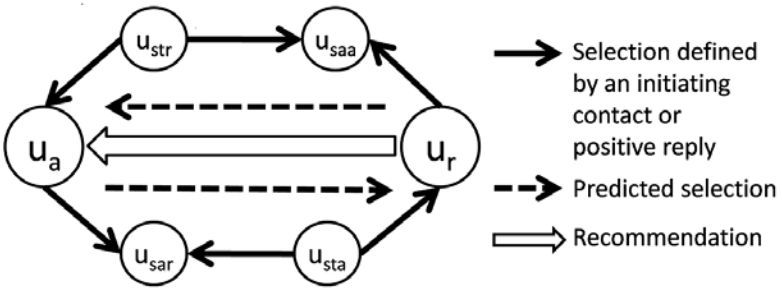


Fig. 1. SocialCollab recommender for bilateral collaborative filtering: u_a is the active user, u_r is the recommended user, u_{str} is a representative user with similar taste to the recommended user, u_{saa} is a representative user with similar attractiveness to the active user, u_{sar} is a representative user with similar attractiveness to the recommended user, and u_{sta} is a representative user with similar taste to the active user

This can be restated in terms of user attractiveness:

3. If u_a likes people with similar attractiveness to u_r, u_a will like u_r;
4. If u_r likes people with similar attractiveness to u_a, u_r will like u_a,

since both assumptions lead to the same predicted selections as illustrated in Figure 1.

Therefore, u_r should be recommended to u_a when u_r likes people with similar attractiveness to u_a and u_a likes people with similar attractiveness to u_r, or equivalently, when people with similar taste to u_r like u_a and people with similar taste to u_a like u_r.

More formally, for a predicted successful interaction between u_a and u_r: denoted $u_a \xrightarrow{*} u_r$, there are two conditions to be fulfilled:

5. The attractiveness of the recommended user should match the taste of the active user, which will facilitate initiation of the interaction from the active user to the recommended user. In a user-based approach, we define this as:

$$u_a \xrightarrow{*} u_r : \exists s, (s \overset{t}{\leftrightarrow} u_a \wedge s \rightarrow u_r) \tag{5}$$

   and its equivalence in terms of predicted selections in an item-based approach:

$$u_a \xrightarrow{*} u_r : \exists s, (s \overset{a}{\leftrightarrow} u_r \wedge u_a \rightarrow s) \tag{6}$$

6. The attractiveness of the active user should also match the taste of the recommended user, to ensure positive responses from the recommended user. In a user-based approach, this can be expressed as:

$$u_r \xrightarrow{*} u_a : \exists s, (s \overset{t}{\leftrightarrow} u_r \wedge s \rightarrow u_a) \tag{7}$$

   and its equivalence in an item-based approach:

$$u_r \xrightarrow{*} u_a : \exists s, (s \overset{a}{\leftrightarrow} u_a \wedge u_r \rightarrow s) \tag{8}$$

Algorithm 1. SocialCollab: Modelling Bilateral Decisions

Initialise $C_{r,a} \leftarrow \emptyset$; $C_{a,r} \leftarrow \emptyset$; $R \leftarrow \emptyset$
find users with similar taste (u_{str}, u_r)
find users with similar attractiveness (u_{sar}, u_r)
for all u_r **do**
  for all $u_s : (u_s \in u_{str}) \wedge (u_s \text{ selects } u_a)$ **do**
    $C_{r,a} \leftarrow C_{r,a} \cup \{u_r\}$ // *users with similar taste to* u_r *who selected* u_a
  end for
end for
for all u_r **do**
  for all $u_s : (u_s \in u_{sar}) \wedge (u_a \text{ selects } u_s)$ **do**
    $C_{a,r} \leftarrow C_{a,r} \cup \{u_r\}$ // *users with similar attractiveness to* u_r *selected by* u_a
  end for
end for
for all $u_r : (u_r \in C_{r,a}) \wedge (u_r \in C_{a,r})$ **do**
  $R \leftarrow R \cup \{u_r\}$ // *recommendation set*
end for
return R

Therefore, we have the following basis for people to people recommendation.

Definition 4. *A recommendation is a predicted successful interaction between two users:*

$$u_r \Rightarrow u_a : (u_r \overset{*}{\rightarrow} u_a \wedge u_r \overset{*}{\leftarrow} u_a) \tag{9}$$

Modelling User Selection We assume that if a user u_1 initiates an interaction by sending a contact to another user u_2, then u_1 likes u_2, which makes sense when considering people's interactions. However, initiating a contact is not the only way people can express their interest in others. If users receive contacts from others, they can also express their interest in the senders by sending positive responses back to the sender. Therefore we extend the model of user selection to include either initiating an interaction or giving a positive response to a contact initiated by another user.

Definition 5. *An extended selection between two users* $(u_i \rightarrow u_j)$ *is a relationship:*

$$u_i \rightarrow u_j : (u_i \rightharpoonup u_j \vee u_i \rightharpoondown u_j) \tag{10}$$

where $u_i \rightharpoonup u_j$ *means* u_i *initiates a contact to* u_j *and* $u_i \rightharpoondown u_j$ *indicates* u_i *responds positively to a contact from* u_j*.*

The SocialCollab Algorithm. As depicted in Algorithm 1, the method works as follows. For each potential recommendation candidate u_r in the dataset, it first finds a set of users u_{str} having similar taste, and another set of users u_{sar} having similar attractiveness, to the candidate u_r. Then u_r is added to the recommendation set R for the active user u_a if at least one similar user in u_{str}

selects u_a and at least one similar user in u_{sar} is selected by u_a. The potential recommendations for user u_a are ranked according to the number of similar users in the set $C_{r,a} \cup C_{a,r}$.

3 Experimental Evaluation

In these experiments we aim to evaluate the proposed approach on people recommendation in a realistic setting. Therefore we applied our algorithm on a social network dataset from a commercial online dating site. We compare our learning algorithm SocialCollab to the standard CF algorithm. Data was pre-processed in Oracle 10 and algorithms were implemented in Matlab.

3.1 Experiment Setup

The datasets were collected from a commercial social network site containing interactions between users. Specifically, the data contains records each of which represents a contact as a tuple containing the identity of the contact's sender, the identity of the contact's receiver and an indicator showing whether the interaction was successful (with a positive response from the receiver to the sender) or unsuccessful (with a negative response).

The experiments were conducted on a training set covering a one week period and a test set on a subsequent week, both in March, 2009. Both training and test sets contain all users with at least one contact in the respective periods. The datasets used are summarised in Table 1.

Table 1. Dataset Description

	#Interactions	#Positive	#Negative	DSR	$\#U_a$ Involved
Training Set	188255	54754	133501	0.29	3746
Test Set	199083	56677	142406	0.28	2865

We compare SocialCollab to the standard CF algorithm using the evaluation metrics defined in the next section.

3.2 Evaluation Metrics

The evaluation metrics used in this research are defined as follows:

Definition 6. *Success Rate (SR) or Precision is the proportion of the true predicted successful interactions to all predicted successful interactions:*

$$SR = \frac{n_{tps}}{n_{ps}}, \tag{11}$$

where n_{tps} is the number of true predicted successful interactions and n_{ps} the number of predicted successful interactions.

Definition 7. *Default Success Rate (DSR) is the proportion of successful interactions to all interactions in the dataset:*

$$DSR = \frac{n_{ts}}{n_{all}}, \tag{12}$$

where n_{ts} is the number of true successful interactions and n_{all} the number of all interactions.

Definition 8. *Success Rate Improvement (SRI) is the ratio of success rate to the default success rate:*

$$SRI = \frac{SR}{DSR}. \tag{13}$$

Definition 9. *Recall is the proportion of the true predicted successful interactions to all true successful interactions:*

$$Recall = \frac{n_{tps}}{n_{ts}}, \tag{14}$$

where n_{tps} is the number of true predicted successful interactions and n_{ts} the number of successful interactions in the dataset.

3.3 Results of Recommendation

We compare SocialCollab to the standard collaborative filtering CF and its extended version CF+ using the proposed selection method defined in Definition 5. More specifically, CF+ uses the extended selection of Definition 5 rather than ordinary selection as used in standard CF. The details of the comparison results of those algorithms on the Top 100 and Top 1000 are shown in Tables 2 and 3, which shows that the proposed algorithms SocialCollab and CF+ both outperform the standard CF for recommendation, with the SocialCollab the best performer. As shown in Table 4, SocialCollab achieves approximately 0.35 SR on average for the Top 100 recommendations for each active user. This gives an SRI of about 1.25. The SRI for CF+ and CF on the Top 100 are less than 1 because the majority of interactions in the dataset are negative, leading to many predicted interactions that are unsuccessful. Figure 2 shows that CF performs at around the default, and CF+ performs better than CF. SocialCollab outperforms CF+.

Table 2. Comparison on SR for Top 100 Recommendations

Top	10	20	30	40	50	60	70	80	90	100
SocialCollab	0.35	0.34	0.34	0.35	0.34	0.35	0.35	0.35	0.35	0.35
CF+	0.30	0.28	0.27	0.27	0.28	0.28	0.27	0.27	0.26	0.26
CF	0.28	0.27	0.26	0.26	0.26	0.26	0.26	0.26	0.25	0.25

Table 3. Comparison on SR for Top 1000 Recommendations

Top	100	200	300	400	500	600	700	800	900	1000
SocialCollab	0.35	0.36	0.36	0.37	0.37	0.37	0.37	0.37	0.37	0.37
CF+	0.26	0.27	0.27	0.28	0.28	0.28	0.28	0.28	0.28	0.28
CF	0.25	0.26	0.26	0.27	0.27	0.27	0.27	0.27	0.27	0.27

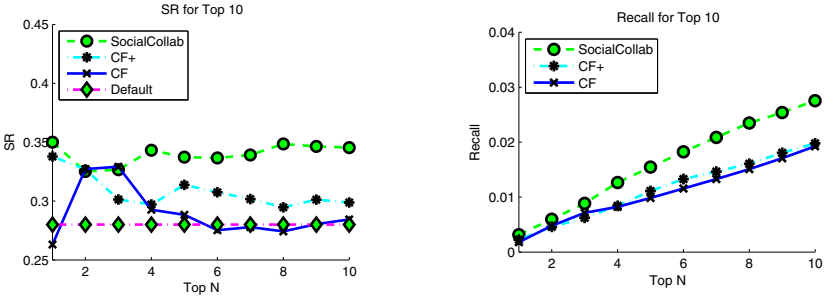


Fig. 2. Comparisons of SR (left) and Recall (right) for Top 10

Table 4. Comparison of Ranked Recommendation Results

		SR		SRI	
		Top 100	Top 10	Top 100	Top 10
A	SocialCollab	0.35	0.35	1.25	1.25
B	CF+	0.26	0.30	0.93	1.07
C	CF	0.25	0.28	0.89	1
D	Default	0.28	0.28	1	1
	Impvt. of A over B	0.09	0.05	0.57	0.18
	Impvt. of A over C	0.10	0.07	0.36	0.25
	Impvt. of A over D	0.07	0.07	0.25	0.25

4 Concluding Remarks

We have proposed an approach for people recommendation by collaborative filtering. Our experimental results show that the novel SocialCollab recommender performs well in people to people recommendation on social network data from a commercial online dating site. The proposed algorithms SocialCollab and CF+ both outperform standard CF as measured on both Precision (SR) and Recall, with SocialCollab being the best. A general framework for ranking in the context of the SocialCollab algorithm is the subject of further work.

Acknowledgement

This project is funded by the Smart Services CRC under the Australian Government's Cooperative Research Centre program.

References

1. Breese, J.S., Heckerman, D., Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 43–52 (1998)
2. Getoor, L., Sahami, M.: Using Probabilistic Relational Models for Collaborative Filtering. In: Working Notes of the KDD 1999 Workshop on Web Usage Analysis and User Profiling (1999)
3. Hofmann, T.: Collaborative Filtering via Gaussian Probabilistic Latent Semantic Analysis. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 259–266 (2003)
4. Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: GroupLens: Applying Collaborative Filtering to Usenet News. Communications of the ACM 40(3), 77–87 (1997)
5. Linden, G., Smith, B., York, J.: Amazon. com Recommendations: Item-to-Item Collaborative Filtering. IEEE Transactions on Internet Computing 7(1), 76–80 (2003)
6. Pavlov, D., Pennock, D.M.: A Maximum Entropy Approach to Collaborative Filtering in Dynamic, Sparse, High-Dimensional Domains. In: Neural Information Processing Systems, pp. 1441–1448 (2002)
7. Shardanand, U., Maes, P.: Social Information Filtering: Algorithms for Automating "Word of Mouth". In: Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 1995), pp. 210–217 (1995)
8. Su, X., Khoshgoftaar, T.: A Survey of Collaborative Filtering Techniques. In: Advances in Artificial Intelligence 2009 (2009)
9. Ungar, L., Foster, D.: Clustering Methods for Collaborative Filtering. In: Proceedings of the AAAI 1998 Workshop on Recommender Systems, pp. 112–125 (1998)

Head X: Customizable Audiovisual Synthesis for a Multi-purpose Virtual Head*

Martin Luerssen, Trent Lewis, and David Powers

Artificial Intelligence Laboratory,
Flinders University
Adelaide, Australia
{martin.luerssen,trent.lewis,david.powers}@flinders.edu.au

Abstract. The development of embodied conversational agents (ECAs) involves a wide range of cutting-edge technologies extending from multimodal perception to reasoning to synthesis. While each is important to a successful outcome, it is the synthesis that has the most immediate impact on the observer. The specific appearance and voice of an embodied conversational agent (ECA) can be decisive factors in meeting its social objectives. In light of this, we have developed an extensively customizable system for synthesizing a virtual talking 3D head. Rather than requiring explicit integration into a codebase, our software runs as a service that can be controlled by any external client, which substantially simplifies its deployment into new applications. We have explored the benefits of this approach across several internal research projects and student exercises as part of a university topic on ECAs.

Keywords: Embodied conversational agents, audiovisual speech synthesis, software library.

1 Introduction

Virtual characters are becoming ubiquitous these days: you are ever more likely to find them in popular movies and TV shows, and their traditional home, the video games business, continues to grow. But while they are becoming increasingly photorealistic, this has no correspondence in our ability to interact with them - even games rely mainly on prerecorded animation and speech. What is missing is an intelligence to act autonomously, the essence of what is referred to as an 'agent'. An anthropomorphic agent has significant potential as a user interface, because interaction with people comes very natural to us. We have years of experience at looking someone in the face and talking to them.

Embodied conversational agents (ECAs) are defined as animated interface agents that engage the user in real-time, multimodal dialogue using speech, gesture, gaze, intonation, and other verbal and nonverbal behaviors that are common to human-human interaction [3]. Even a basic implementation of such an

* This work was funded by the Thinking Head project, an Australian Joint ARC/NHMRC Thinking Systems Special Research Initiative.

J. Li (Ed.): AI 2010, LNAI 6464, pp. 486–495, 2010.

agent must embrace a variety of technologies in the field of artificial intelligence - including computer vision, speech recognition, and dialogue management. However, it is the synthesis that makes the first and lasting impression. After all, if it doesn't look or sound like you can or want to have a conversation with it, you won't.

Animating a character is hardly unique, but ECAs present special challenges here, as they need to respond in real-time to events and should be able to say what they want - i.e. prerecording is often not an option. The synthesis also needs to fit into a larger architecture of ECA components that it interacts with. In light of these objectives, we present *Head X*, an audiovisual synthesis platform for ECAs that offers extensive configurability and convenient interoperability with the many other sides of ECA research. The remainder of this paper will detail the technical design and capabilities of the component as well as our current work towards applying it to research and education.

2 Background

Several research groups have previously devised noteworthy ECA synthesis systems. The virtual head Baldi can synthesize highly realistic speech movements and has found principal use as a speech teacher [7]. Indeed, many animated agents are intended for pedagogical use, although they typically offer only basic synthesis and interactivity [10]. Fewer agents also incorporate perceptual and reasoning capabilities, such as REA, MIT Media Lab's virtual real estate agent [1]. Less task specific and more targeted at studying interhuman communication are the virtual body Greta [12] and virtual head RUTH [11], although the emphasis here is firmly on synthesis. Virtual body Max straddles a middle ground, adding gestural interaction in a virtual environment [6].

While each of the above agents has substantial research value in its own right, each also has a specific, unchanging identity. This may significantly limit their reusability in other applications, because users tend to be quite particular about whom they wish to interact with [5]. It has also been our observation that a character's appearance and voice can easily overshadow any other qualities of the actual interaction. Head X was therefore developed with diversity in mind: diversity in what it can synthesize, but also in what it can be used for. As the face is generally the most expressive part of the body, our software focuses on synthesizing a head. The next section will describe the basic technology involved in this.

3 System Design

3.1 Face Synthesis

Creating a virtual 3D face is typically the job of an artist, who would invest substantial manual effort into making it look natural and real. This is rarely

practical if your budget is small, you need many faces, or have frequently changing requirements. In such circumstances, it is better to derive faces from a 3D parametric model whose parameters intuitively lead to the desired outcomes, so that even non-artists can produce realistic faces. A well-established process for obtaining such a model is to scan different faces of real people with a laser [2]. The densely sampled geometry that results from this must be matched point-to-point with the geometry of all other faces, so that a statistical model can be fitted to the face distribution. Principal component analysis (PCA) can extract the most important face parameters, but to relate these to facial attributes such as gender or age, faces need to be labeled to indicate the markedness of an attribute. The same process can also be followed to distinguish facial expressions.

To save on effort, quality parametric models can be purchased. We acquired an academic license to the FaceGen SDK, a parametric face model that also includes blendshapes for facial expressions [4]. As illustrated in Fig. 1, FaceGen's 'PhotoFit' systems also allows a 2D photo of a person to be converted into a 3D model semi-automatically. Use is limited to those universities with a license, but it is a comparatively affordable option. Alternatively, we allow the user to import an existing 3D model and all the needed blendshapes manually; this allows for potentially greater variety (and the addition of accessory models, such as hair and glasses), as well as freedom from licensing. Head X visualizes the face mesh in real-time using OpenGL. Users can customize the pixel shaders to change the surface properties of objects. Face meshes and textures can also be morphed while the user is interacting with the software, as shown in Fig. 2.

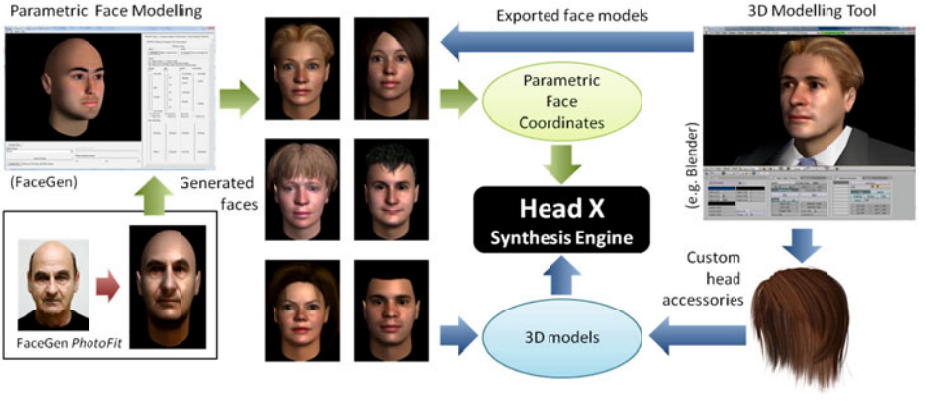


Fig. 1. Head X animates a virtual head using a FaceGen 3D model and a corresponding face coordinate obtained from the freely available FaceGen Modeller, which also supports the fitting of a 2D picture to the model. Alternatively, the face can be animated from a set of models, one for each desired blendshape, built with any basic 3D modelling tool. Head accessories, such as hair, glasses, and jewelry, can also be added in this way.

Fig. 2. Head X offers a diversity of faces and also supports face morphing, even without a parametric face model. Face shape and texture can be changed interactively while speech and expressions are animated.

3.2 Speech Synthesis

In video games, animated characters are typically voiced by humans, and the animation must simply follow the audio signal. Conversely, an ECA should be able to speak any text. A variety of software options exist for synthesizing the audio component of speech, but they are rarely as freely configurable as the faces above. Natural sounding voices mostly follow a unit selection approach that picks speech fragments from a large database of speech of a single person. Because of the effort involved in collecting this data, the best voices tend to be offered commercially. Alternatively, a more direct synthesis of speech, such as diphone synthesis, allows for greater customization, but at a cost to the naturalness of the voice. Head X supports two text-to-speech (TTS) interfaces, Microsoft's Speech Application Programming Interface (SAPI) and DFKI's Modular Architecture for Research on speech sYnthesis (MARY).

The SAPI 5.3 TTS interface [8] is used by a vast library of commercial voices in multiple languages and accents (indeed, we have partnered with groups from Germany and China on developing multilingual heads). Additionally, SAPI supports XML extensions, including SSML, that can be added to a text to modify rate, pitch, and other aspects of the speech. The MARY v4.0 TTS system [14] also supports custom XML extensions for changing the voice, although fewer voices are available. However, the MARY TTS and MARY voices are open source, which make them highly suitable for research purposes. MARY supports both unit selection, diphone, and hidden Markov model based voice synthesis and includes tools for building your own voice.

3.3 Speech Animation

The FaceGen model comes with a comprehensive set of mouth shapes used during speech (so-called visemes), which need to be animated in synchrony with the audio output. Rather than analyzing the text or audio, we can use the SAPI and the MARY TTS systems directly for this. The SAPI interface provides real-time feedback from the voice engine. Viseme events are triggered whenever a viseme boundary is reached. The SAPI 5 viseme set consists of 22 visemes that are based on the well-known 'Disney 13' set. Each SAPI event contains the current

viseme code (SP_VISEME_...) and its duration. The absolute position of the viseme can be determined from the position in the audio stream when the event was triggered. We tested several commercial voices by Microsoft, Cepstral, and Nuance and found that while they all trigger viseme events, the visemes differ substantially between engines and voices even. In particular, the tested Nuance voices deliver multiple events for the same viseme code, albeit at sequential and sensible temporal locations and durations. It can be difficult to interpret this correctly; our system therefore combines sequential events of the same viseme code into one event of larger duration. This is seen as step 2 in Fig. 3(a) where second and third events labelled "v2" are merged into a single"v2" event.

MARY does not support viseme events, but we can retrieve timed SAMPA phonemes for the outputted speech. To receive the sequence and duration from the MARY TTS the REALISED_DURATION output from the MARY server is utilized. This returns a duration and the SAMPA(-like) phoneme. The absolute position is calculated as the cumulative of the previous durations.

Both the MARY phonemes and the SAPI visemes need to be mapped to the blendshapes of the 3D model - neither exactly matches what is provided by FaceGen. We have established visually acceptable mappings for each, based on earlier work by Wang et al. [15]. For custom 3D models, the mapping can be separately modified via XML configuration files.

The selected sequence of blendshapes then needs to be blended into an animation. For this purpose, we take a user-defined number of frames per second (30, by default) and weigh all the nearby blendshapes discretely for each frame. The weighting is derived from the mapping itself (if two or more morphs are specified), a user-modifiable amplitude variable, and the current ramp value. As illustration in step 4 of Fig. 3(a) each viseme or morph is transitioned to and from by increasing and decreasing (or ramping) the weight associated with that morph's frame. The style of ramp is controlled by a customizable ramping function. The ramping extends to a user-modifiable overlap into the adjacent frames. To avoid multiples of the same morph being present in one rendered frame, the maximum weight of any repeated frames is taken. Even a smooth ramp and overlap is not safe from aliasing artifacts, however. To overcome this a final mixing of the current and adjacent frames is performed (as depicted in Fig. 3(b)).

Non-verbal Animation. Speech is not the only means by which the Head can communicate with the user. Head X allows the user to define composites of all blendshapes, including those expressing emotions or eye brow/lid movements, as expressions that can be triggered interactively (see Figure 4 for the big six emotions). Note that when multiple blendshapes are applied simultaneously, unwanted results may appear because the blendshapes will stack. Our system is therefore designed to offer multiple expression channels, where expressions on separate channels only interact in restricted ways. By default, there is a speech channel, an emotion channel, and an idle channel; expressions do not stack between these. The idle channel is automatically filled with randomly scheduled animations, such as head bobbing and blinking and subtle expressions that give

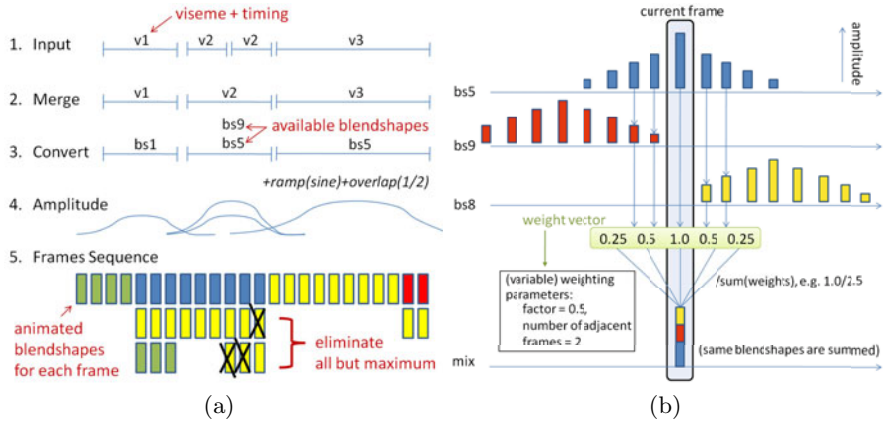


Fig. 3. Viseme animation: a) Five steps to animating visemes across image frames: superfluous identical visemes are merged, then mapped to the available blendshapes via a customizable ramping function, and finally discretized into frames; b) mixing of the current and adjacent frames to improve aliasing artifacts

Fig. 4. Head X offers a diversity of expressions shown here are the "Big Six" emotions: happiness, sadness, anger, surprise, fear and disgust

the viewer the impression that the head is alive - this can be freely configured and disabled and can be used to enable non-verbal backchannelling in dialogue scenarios.

3.4 External Interfaces

Rather than being an objective in its own right, Head X is intended to work with other technologies that could benefit from embodiment. For instance, a head tracker becomes a head that follows the user, thus greatly increasing the perceived impact of the tracking technology. However, we want researchers and other developers to spend time on their applications, not on Head X. In fact, our design does not require the developer to read or compile a single line of Head X code. Head X runs as a service that fulfills requests by one or more other programs running concurrently, which are permitted full access to both high-level (e.g. say this! look there!) and low-level features (e.g. animation blend shapes).

The development process remains very simple if you only need basic features and gradually becomes more complex as you acquire more control. Satisfying this condition with a single interface would have been intractable, so a variety of interfaces are supplied.

Command Processor. It is important to balance power and diversity against unnecessary complexity, so a certain homology to the Head X interfaces had to be established. All action requests - so-called commands - are centrally processed by the command processor, including those arising from interaction with the graphical user interface, or any other interface. Commands are scheduled and processed in sequence. It does not matter where the command originated, the outcome remains consistent.

By default, text sent to Head X is interpreted as something to be said, not as a command. Commands can be included inside text, demarcated by curly brackets, as angled brackets can cause problems with certain XML parsers of external components. These commands are triggered when their in-text location is reached during speech output. We can change facial expressions, voices, backgrounds - indeed, any modifiable part of Head X - interactively in this fashion. Text can be sent either from a subwindow of the user interface, an on-screen OpenGL prompt, or remotely via messages or shared memory (see also Fig. 5).

Configuration. An XML configuration file holds the base settings of Head X, including 3D models to load, animation and graphics parameters, voice selection and viseme-to-blendshape mapping, idle behaviour, composite expression definitions, window properties, subtitle automation, input modes, and which memory sharing interfaces to expose. Head X can thus be customized to suit a variety of needs. Full

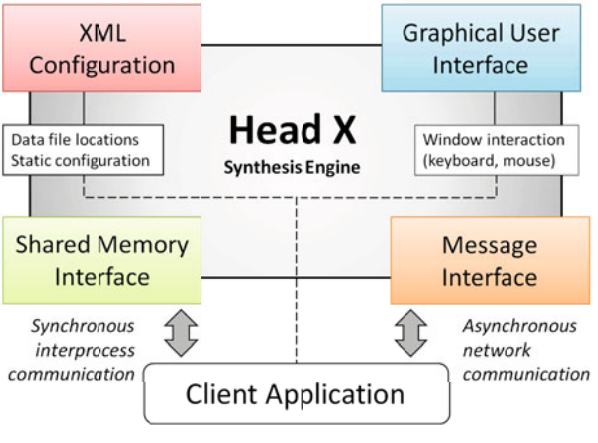


Fig. 5. The interfaces of Head X: remote control by a client application is mainly achieved through shared memory and/or network messaging, although properties of the configuration and GUI can be passed on and manipulated as well

or partial XML configurations can be loaded from a file or an in-line string using commands anytime, even during dialogue. For most parameter changes, response time is instantaneous, so changes can be applied on the fly.

Messages. To control Head X from within your own separate application, you can send messages via the TCP protocol, even across a network. The basic message type simply includes a text to be parsed and spoken. Further message types can signal when speech has ended or some other action is triggered or completed. Clients that link with our lightweight message library (available in C++ and Java) can sign up to events so that callbacks are performed whenever a message of a particular type arrives. A web client is also available for observing messages remotely.

Shared Memory. While message passing can satisfy most basic requirements, many applications and projects need more direct control. In particular, our messaging system does not guarantee synchrony and also creates substantial load if thousands of messages are passed every second, such as when driving the animation remotely. Consequently, a second external interface is provided that relies on shared memory and process synchronization, which is very close in performance and control to actually integrating Head X into your application directly. Unlike with messages, it will only work for processes on the same machine. The Head X configuration specifies what data objects should be exposed publicly, so that other processes can observe and possibly modify the data. Text can be sent to the head this way, but you can also observe the status of the head; what and whether something is being spoken or loaded; where we are in a sentence; mouse and keyboard inputs; and also modify animation morphs directly and change camera position, lighting, and subtitles. Client libraries for the shared memory interface are available in C++, Java, and C#.

4 Applications

We are currently employing Head X for several ongoing research projects that involve the use of virtual characters in teaching and counseling. They include a training exercise for autistic children, for which the virtually character acts as a presenter [9], and a talking calendar for elderly people, which links Google Calendar to a virtual character and allows speech interaction in both directions [13]. Developing these as external clients of Head X has substantially accelerated our progress, because the involved researchers only need limited technical knowledge of Head X, which permits them to focus on their own areas of expertise and favorite programming languages. Since our messaging and shared memory interfaces are not exclusive to Head X, we have also been able to build up a library of components that can be shared within a larger ECA framework, allowing researchers to benefit from prior work; including video processing pipelines using the shared memory interface, audio-visual speaker association and recognition, speech and emotion recognition, user tracking, evolutionary learning, and dialogue management.

It is not just researchers who benefit, however; it is also the students, who are directly exposed to a new way of interacting with a computer. We are presently employing Head X for teaching a 4th-year university topic on ECAs, where students independently develop new ECA applications as part of their project work. Although it has only been run once so far, we have had some interesting outcomes, including a singing head, a person tracker, a music database, and a speech-triggered, multi-agent card game. The opportunity to develop a talking head application adds greatly to the applied learning experience of the course.

The service-based architecture of Head X combined with comprehensive documentation and sample programs has ensured that students rarely find it too challenging to develop something. The main issues appear to lie in synchronizing the application with what the head is doing at any given time (e.g. not to interrupt ongoing speech). We expect to address this by offering higher-level helper methods for the client interfaces. Another feature that has been frequently requested and is under consideration is the ability to animate and interact with other 3D models (not just head accessories) in the space of the head, as application proposals rather quickly grow from virtual heads to entire virtual worlds.

5 Conclusions

Head X synthesizes a talking head whose face and voice can be freely customized to suit most needs. This encourages its use in applications that could benefit from an anthropomorphic interface, because you can always create a character that suits the task at hand. Integration is simplified by having the software controlled remotely across a network or by memory sharing, so no linkage or compilation of Head X code is needed. We have explored the benefits of this approach in research and student projects, with diverse and interesting applications being built around Head X. One of the remaining concerns is that the commercial parametric face model limits use to licensed institutions. We intend to make publicly available a completely free version[1] of Head X in the near future that will still allow you to import and morph faces from other modeling tools. Furthermore, ECA applications can rapidly become very complex, as synthesis is only one of many components. We will hence continue to work towards establishing a more general framework that allows such situations to be handled robustly and easily, so that developers can focus on that ultimate goal of making human-computer interaction more human.

References

1. Bickmore, T., Cassell, J.: Social dialogue with embodied conversational agents. In: van Kuppevelt, J., Dybkjaer, L., Bernsen, N. (eds.) Advances in Natural, Multimodal Dialogue Systems, pp. 23–54. Kluwer Academic, New York (2005)

[1] http://csem.flinders.edu.au/research/programs/th/projects/

2. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: SIGGRAPH 1999: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 187–194. ACM Press/Addison-Wesley Publishing Co., New York (1999)
3. Cassell, J.: Embodied conversational agents: representation and intelligence in user interface. AI Magazine 22(3), 67–83 (2001)
4. FaceGen SDK 3.6, http://www.facegen.com
5. Gulz, A., Haakeb, M.: Design of animated pedagogical agents a look at their look. Int. J. Human-Computer Studies 64, 322–339 (2006)
6. Kopp, S., Sowa, T., Wachsmuth, I.: Imitation games with an artificial agents: From mimicking to understanding shape-related iconic gestures. In: Braffort, A., Gherbi, R., Gibet, S., Richardson, J., Teil, D. (eds.) Gesture-Based Communication in Human-Computer Interaction, pp. 436–447. Springer, Berlin (2004)
7. Massaro, D.W.: From multisensory integration to talking heads and language learning. In: Calvert, G., Spence, C., Stein, B.E. (eds.) Advances in Natural, Multimodal Dialogue Systems, pp. 153–176. MIT Press, Cambridge (2004)
8. Microsoft Speech API 5.3, http://msdn.microsoft.com/speech
9. Milne, M., Luerssen, M., Lewis, T., Leibbrandt, R., Powers, D.: Development of a virtual agent based social tutor for children with autism spectrum disorders. In: Proc. 20th Int. Joint Conf. on Neural Networks, pp. 1555–1563. IEEE, Los Alamitos (2010)
10. Moreno, R., Flowerday, T.: Students' choice of animated pedagogical agents in science learning: A test of the similarity-attraction hypothesis on gender and ethnicity. Contemporary Educational Psychology 31, 186–207 (2006)
11. Oh, I., Stone, M.: Understanding RUTH: Creating believable behaviors for a virtual human under uncertainty. In: Duffy, V.G. (ed.) HCII 2007 and DHM 2007. LNCS, vol. 4561, pp. 443–452. Springer, Heidelberg (2007)
12. Poggi, I., Pelachaud, C., de Rosis, F., Carofiglio, V., Carolis, B.D.: GRETA. A Believable Embodied Conversational Agent. In: Stock, O., Zancarano, M. (eds.) Multimodal Intelligent Information Presentation, vol. 27, pp. 3–25. Springer, Netherlands (2005)
13. Powers, D., Luerssen, M., Lewis, T., Leibbrandt, R., Milne, M., Pashalis, J., Treharne, K.: MANA for the Aging. In: Proceedings of the 2010 Workshop on Companionable Dialogue Systems, ACL 2010, pp. 7–12. ACL (2010)
14. Schroder, M., Trouvain, J.: The German text-to-speech synthesis system MARY: A tool for research, development and teaching. International Journal of Speech Technology 6(4), 365–377 (2003)
15. Wang, A., Emmi, M., Faloutsos, P.: Assembling an expressive facial animation system. In: Sandbox 2007: Proceedings of the 2007 ACM SIGGRAPH Symposium on Video Games, pp. 21–26. ACM, New York (2007)

Diagnosing Component Interaction Errors from Abstract Event Traces

Wolfgang Mayer, Xavier Pucel, and Markus Stumptner

University of South Australia, Adelaide, SA, 5095, Australia
{mayer,xavier.pucel,mst}@cs.unisa.edu.au

Abstract. While discrete event systems have been widely applied for diagnosing distributed communicating systems, existing models may not completely satisfy the requirements for the application of fault identification and repair in software systems. This paper presents a model-based diagnosis approach that identifies possible faults based on generic fault models in abstract traces where events may be associated to multiple system components. We overcome the common limitation that precise fault models are available for each component and leverage generic fault models of classes of faults instead. We show that diagnoses representing entire classes of equivalent solutions can be computed based on local information and investigate the performance of our algorithm.

1 Introduction

The complexity and size of software systems have rapidly increased in recent years, with software engineers facing ever growing challenges in building and maintaining such systems. In particular, *testing and debugging* still constitutes a major challenge in practice, as demonstrated by the many research projects dedicated to this topic [2,6] and the sheer number of software faults identified by commercial providers [3].

Since testing and debugging are among the most costly and time consuming tasks, a variety of intelligent debugging aids have been proposed within the last three decades. Model-based software debugging (MBSD) [9,8] is a particular technique that exploits discrepancies between a program execution and the intended behaviour to isolate program fragments that could potentially explain an observed misbehaviour. However, most model-based techniques have been limited to single programs and have assumed complete observability of program states. The shift from single programs to distributed systems requires adaptation of debugging and diagnosis approaches.

This paper extends previous work on diagnosing failed program execution traces to distributed systems where multiple communicating software components are observed. From models of the correct interface protocol of each component potential faults that may explain the events observed in an execution are inferred. We build a system model from a set of transition systems and introduce fault transitions that reflect generic classes of faults. Every alteration of the system can be described in terms of our generic classes, which allows us to deal with totally unforeseen faults. The approach is not limited to the debugging of programs but also applies to the general class of discrete event systems.

[1] This research is partially funded by the Australian Research Council (Grant DP0881854).

J. Li (Ed.): AI 2010, LNAI 6464, pp. 496–505, 2010.

The contributions in this paper are as follows: (i) We provide a formal diagnosis model based on discrete event systems under partial observability. Different from classic assumptions our model utilizes generic fault transitions that reflect *classes* of faults rather than specific fault transitions associated with an automaton. (ii) Our model also allows for ambiguity in events in that an observed event may correspond to a number of events stemming from different automata in the model. (iii) We develop a diagnosis algorithm that computes the set of minimal explanations incrementally by minimally unfolding the global transition relation based on an observed event sequence. The algorithm computes classes of equivalent solutions rather than identify all equivalent solutions. Each solution corresponds to a repair modification in the abstract system model. This paper is organised as follows: We proceed by motivating our approach on a simple example in Section 2. The formal diagnosis model is described in Section 3. Its properties and our diagnosis algorithm is discussed in Section 4. The performance of the approach is assessed empirically in Section 5. Related work is discussed in Section 6 before summarizing the contributions in this paper in Section 7.

2 Motivating Example

The general problem context considered in this work is the diagnosis of communicating systems where only some events can be observed while others are hidden. We assume that the for each system component an automaton specifying its correct behaviour is available. Each automaton specifies the expected component behaviour in terms of event sequences. Events in this context correspond to executions of particular program instructions or messages exchanged between subsystems. Some of these events can be observed in an execution trace, for example because relevant debugging instrumentation has been furnished, while other events remain unobserved. While our model assumes all observable events that are generated appear in the trace, the information attached to each event may not be sufficient to associate an event with the emitting component.

Let us consider for example a software system composed of two parallel threads that execute the same program, which reads from and writes to some data structure protected by a shared lock. The model of the two threads and the lock are depicted in Fig. 1. The behavior specified by the automata is implemented in a program that we have instrumented in order to observe read and write accesses to the data, and lock acquisitions and releases. In this model, events l_i and u_i represent respectively the acquisition and release of the lock by thread i. They occur synchronously in all relevant automata. In contrast, events r and w represent read and write access to the data, which are not synchronized. When observing an r event, it is impossible to determine which thread actually executed it. For the sake of clarity, in this example all events are observed. As we will see later, our approach also copes with unobserved events, and in particular unobserved synchronous events.

For example, assume that the event sequence given on the right hand side in Fig. 1 has been observed. The sequence conflicts with the models of the threads and the lock, since unprotected r and w events occur between the u and the subsequent l event. Using only the models of the normal system behaviour, no precise explanation can be devised: since both threads and the lock are required to derive a conflict with the trace, all components are possible explanations.

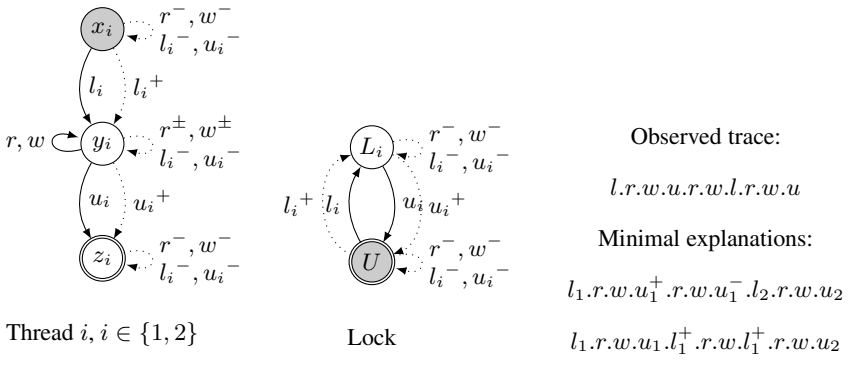


Fig. 1. Model of three software components: two threads accessing data, a mutual exclusion lock, and an observed event trace. Events r and w are autonomous, and l_i and u_i are shared. Fault transitions are dotted. All non-fault events are observable. Initial states are shaded, final states are double-stroked.

We complement the nominal system behaviour with generic fault transitions that reflect the classes of faults that manifest as extraneous events (e^-) or as missing events (e^+) in the observed trace. The fault transitions are shown as dotted arcs in Fig. 1, and labels reflect which events are extraneous or missing. These fault models are created automatically from the normal system models as described in Section 3.

The fault models allow us to identify the minimal alterations that must be made to the trace to resolve the conflict with the system models. In this example there are two minimal explanations (as shown on the right hand side in Fig. 1): the unlock operation before the conflicting data access could be removed (denoted by fault event ${u_1}^-$) and a corresponding lock acquisition operation could be inserted after the access (fault event ${l_2}^+$); or additional lock and unlock operations (${l_1}^+$ and ${u_1}^+$) could be inserted before and after the unprotected data access. From these diagnoses manual investigation or further debugging techniques on the source code level could be applied in order to correct the problem.

Our diagnosis procedure can identify entire sequences of events that must be inserted and removed in order to explain an observed discrepancy. In addition to simple symmetries that may arise from ambiguity in the association of events to system components, further equivalent solutions where fault events are possible at one of multiple points in an observed trace are omitted. As a result, the number of explanations focuses on different resolutions to the problem rather than enumerate all possible fault assumptions.

3 Diagnosis Model

Our diagnosis model is based on the principle of consistency-based diagnosis [11], where a model of the nominal behavior of a system is contrasted with the actual behavior exhibited by a system. Discrepancies between the observed behavior and the behavior predicted by the model can be exploited to infer possible behavioral changes ("diagnoses") in the model that may explain the differences. The diagnosis model

presented in this paper adopts the principles of Discrete Event Systems (DES) [5] to describe the nominal behavior of a system.

Definition 1 (Labeled Transition System, LTS). *A labeled transition system is a tuple* (S, s_0, F, E, T) *where* S *is a set of states,* s_0 *is the initial state,* $F \subseteq S$ *is the set of distinguished final states,* E *is a finite set of transition labels ("events"), and* $T \subseteq S \times E \times S$ *is the set of labeled transitions. We write* $s \xrightarrow{a}_T s'$ *for* $(s, a, s') \in T$. *We omit the subscript* T *if it is clear from the context.*

We specify the nominal behavior of a system using a set of LTSs. Each LTS corresponds to a system component and governs the sequences of events that are considered correct. Such discrete event models are particularly well-suited to specifying the expected behavior and possible interactions between components without the need to consider any particular implementation of components. We will use the terms *component* and *LTS* interchangeably if there is no ambiguity.

Definition 2 (System Model). *A system model* $\left(Cs, E, E^S, E^O\right)$ *(describing the system's nominal behavior) is based on a set of labeled transition systems (LTS)* $Cs = \left\{C^1, \ldots, C^n\right\}$, *where each* C^i *is an LTS* $\left(S^i, s_0^i, F^i, E^i, T^i\right)$. *The transition relation* T^i *in each LTS specifies the possible event sequences that can be exhibited by an individual component* C^i *if* C^i *is correct. Let* $E = \bigcup E^i$ *denote the set of events present in the system model.* E *can be partitioned into* shared events $E^S \subseteq E$ *and* autonomous events *(*$E \setminus E^S$*). Some events are* observable *(*$E^O \subseteq E$*).*

The behavior of the entire system is obtained from the system model by linking the individual LTSs based on the shared events in E^S. The evolution of the system is constrained such that transitions labeled with a shared event e must occur simultaneously in all components where e appears

Definition 3 (Synchronous Product). *Let* $A_i = \left(S^i, s_0^i, F^i, E^i, T^i\right)$, $i \in \{1, 2\}$ *be two LTS with shared events* E^S. *Let* $\sim \subseteq E^1 \times E^2$ *be an equivalence relation between events. The synchronous product transition system* $A_1 || \underset{E^S}{\sim} A_2$ *is defined as the LTS* (S, s_0, F, E, T) *where* $S = S^1 \times S^2$, $s_0 = \left(s_0^1, s_0^2\right)$, $E = E^1 \cup E^2$, $F = F^1 \times F^2$, $\left(s^1, s^2\right) \xrightarrow{e}_T \left(s'^1, s'^2\right)$ *if and only if (i):* $e \in E^1 \setminus E^S, s^1 \xrightarrow{e}_{T^1} s'^1$ *and* $s'^2 = s^2$, *(ii):* $e \in E^2 \setminus E^S, s^2 \xrightarrow{e}_{T^2} s'^2$ *and* $s'^1 = s^1$, *or (iii):* $e, e' \in E^S, e \sim e', s^1 \xrightarrow{e}_{T^1} s'^1, s^2 \xrightarrow{e'}_{T^2} s'^2$. *The synchronous product operation is commutative and associative. For brevity we will assume that* $\sim = \{(e, e) \mid e \in E\}$ *unless noted otherwise.*

The synchronous product may contain states that are not reachable from the initial state, or states from which no final state is reachable. In an abuse of notation, we do not distinguish the synchronous product and the automaton obtained by removing these states and the transitions related to them.

The synchronous product of all LTS in a system model, called the *global model* $G = C^1 || \underset{E^S}{\sim} \ldots || \underset{E^S}{\sim} C^n$, represents all valid event sequences that a correct system may exhibit. While this model is convenient to *define* the diagnosis problem addressed in this paper, the size of G prohibits its explicit construction for all but trivial systems.

Definition 4 (Trace). *A* trace t *of events is a finite sequence* $e_1.e_2.\ldots.e_n$ *of events from* $e_i \in E$. *Let* ϵ *denote the empty trace. For each trace there exists an LTS* $R = (V, v_0, \{v_n\}, U, W)$ *where* $U = \{e_1, \ldots, e_n\}$ *and* $W = \left\{v_{i-1} \xrightarrow{e_i} v_i \mid i \in \{1, \ldots, n\}\right\}$.

For brevity we do not distinguish between the trace and its corresponding LTS. A trace is accepted *by an LTS A if and only if a final state in $A||^{\sim}_{E\cup U}R$ is reachable from (v_0, s_0).*

A trace reflects correct behavior if it is accepted by the global model G. However, not all events in an execution can be observed. Recall that the events in G are partitioned into observable events, E^O, and unobservable events. Therefore, an observed trace t reflects a correct execution if there is a trace t' accepted by G that exhibits the same sequence of observable events. The problem of deciding whether an observed trace is indeed accepted by G is further complicated by the fact that an autonomous event e may occur in different transition systems C^i.

If an observed trace t is not accepted by G, a fault must have occurred in the execution. Faults in a system's implementation can manifest themselves as extraneous events or as missing events in a trace. In order to isolate particular faults, the system model is amended with additional transitions that reflect extraneous events in a trace and events that have been omitted from the trace. The presence of an extraneous event e in a trace can be modeled by adding a transition to the system model that consumes e but does not change the state. Similarly, the absence of an event e from the trace can be modeled by duplicating and relabeling an existing transition for e.

Definition 5 (System Fault Model). *Let $M = \left(Cs, E, E^S, E^O\right)$ with $Cs = \left(C^1, \ldots, C^n\right)$ be a system model with $C^i = \left(S^i, s_0^i, F^i, E^i, T^i\right)$ and let G represent its global model. The System Fault Model M_F with LTSs $\left(C_F^1, \ldots, C_F^n\right)$ and $C_F^i = \left(S^i, s_0^i, F^i, E_F^i, T_F^i\right)$ is obtained from M by amending E^i and T^i to include unobserved fault events e^- and e^+ for all $e \in E$. A transition labeled e^- represents the fault where event e is present in the trace but is not admitted by G, whereas a transition labeled e^+ represents the absence of an event e in the trace:*

$$E_F^i = E^i \cup \left\{e^-, e^+ \mid e \in E^O\right\},$$
$$T_F^i = T^i \cup \left\{ s^i \xrightarrow{e^-} s^i \;\middle|\; s^i \in S^i, e \in E^O \right\} \cup \left\{ s^i \xrightarrow{e^+} s'^i \;\middle|\; s^i \xrightarrow{e}_{T^i} s'^i \right\}.$$

E^S is amended correspondingly.

A global model of the system including all possible faults can be obtained by building the synchronized product from the individual C_F^i. This global fault model represents all possible evolutions of the system, including normal and faulty behavior. We will use G_F to refer to the global system model derived from a given diagnosis problem M_F.

The diagnosis problem can now be defined by fixing the system fault model and an observed execution trace:

Definition 6 (Diagnosis Problem). *A diagnosis problem is a tuple (M_F, r) where $M_F = \left(Cs, E, E^S, E^O\right)$ is a system fault model and r is an event trace over E^O.*

A diagnosis Δ for a diagnosis problem is an event trace that is accepted by G_F.

Definition 7 (Diagnosis). *Let $P = (M_F, r)$ be a diagnosis problem with associated global model G_F, and let E^O be the set of observable events in G_F. Relation $\sim_F$ extends $\sim$ to match non-fault events in r with fault events in G_F: $\sim_F = \sim$*

$\cup \left\{(e, e'), (e', e) \mid e \in E^O, e' \in \{e^+, e^-\}\right\}$. *A trace* δ *is a* diagnosis *for* P *if and only if* δ *is accepted by* $G_F||_{E^S}^{\sim_F} r$.

Let E^F *be the set of all fault events in* M_F*, and let* $\preccurlyeq \subseteq E \times E$ *be a partial ordering of events where all non-fault events are preferred to fault events:* $e \in E \setminus E^F, e' \in E^F$, $\epsilon \preccurlyeq e \preccurlyeq e'$. *Relation* $\preccurlyeq$ *can be extended to an order on event traces such that* $\sigma^1 \preccurlyeq \sigma^2$ *implies* $\sigma^1\sigma \preccurlyeq \sigma^2\sigma$ *and* $\sigma\sigma^1 \preccurlyeq \sigma\sigma^2$. *A diagnosis* δ *is* minimal *if there is no diagnosis* $\delta' \preccurlyeq \delta$.

Different ordering relations $\preccurlyeq$ can be used. For example, $\preccurlyeq$ can be defined to dynamically adjust event orderings based on the previous events in a trace to account for faults that appear multiple times in a trace or to model dependent or context-specific faults. In this paper, we focus on a static ordering where all non-fault events are preferred to any fault event. We do not order different fault events, but consider their frequency; traces with fewer fault events are preferred.

4 Diagnosis Computation

The global model is convenient to characterize the diagnosis problem and its solutions, but it does not allow us to efficiently compute diagnoses. We present an incremental approach to enumerating diagnoses that avoid constructing the global system model. Discrepancies between the observed trace and individual LTSs in the system model are exploited to extend the trace with fault transitions that may resolve the discrepancy. Equivalent diagnoses are identified from symmetries in the model and pruned.

The algorithm we present interleaves the computation of reachable states in the local models and the global model with the association of events to system components. A key element of our approach is that much of the global model can be ignored when computing diagnoses for a given observed trace.

For a given diagnosis problem (M_F, r) diagnoses can be constructed incrementally guided by the observed events in r. The idea is to minimally extend the set of partial explanations ("prefixes") to account for the next event in the observed trace. We prune execution paths that are inconsistent with the observed trace or redundant because they describe equivalent thread interleavings. By aligning the diagnosis construction with the actual observations, only the relevant states of G_F will be visited. Starting at the initial state of each LTS C^i, a prefix σ of a diagnosis can be computed incrementally.

We show that prefixes can be incrementally constructed based on local information without building the global model explicitly (Theorem 1), that enabled fault transitions partition possible explanations into equivalence classes (Theorem 2), and that symmetries in the problem can be reduced by restricting the synchronized product operation with a static ordering of components (Theorem 3).

Definition 8 (Diagnosis Prefix). *A event trace* σ *is a prefix of a diagnosis* δ *if there is an event sequence* δ' *such that* $\delta = \sigma\delta'$.

Since δ is a diagnosis for r, the sub-sequence of δ that contains only events in E^O is also a prefix of r: the prefix partitions r into the events that have already been accounted for (r_σ) and the remaining events ($r_{\delta'}$) in the trace. This observation can be exploited to limit the possible extensions of a given prefix σ.

Theorem 1. *Let $s = \left(s^1, \ldots, s^n\right)$ be a state in G_F and let e be the next observed event in r. It is sufficient to consider the paths p^i in C^i from s^i to a transition labeled with $e' \sim_F e$ to compute all relevant extensions $\sigma' = p^i||_{E^S}^{\sim} \ldots ||_{E^S}^{\sim} p^n$ of σ in G_F. Only unobservable transitions and fault transitions need to be considered in each p^i.*

Proof. In order to consume e, it is necessary to find an enabled transition in G_F that is labeled with an event $e' \sim_F e$. Such a transition may be immediately available in s, or may become available after one or more transitions representing fault events are traversed. Therefore, it is sufficient to consider only the paths in C^i that originate in the i^{th} state component of s. If an observable non-fault transitions labeled e'' occurred in σ' it would be observed in r. Since $e \not\sim_F e''$ such a path is infeasible and need not be considered.

Fault assumptions that are unnecessary to enable subsequent fault transitions or the observed event transition for the same diagnosis prefix can also be omitted.

Theorem 2. *Let $\sigma = \langle s_1 \ldots s_j t_j s_{j+1} \ldots s_m \rangle$ and $\sigma' = \langle s_1 \ldots s_j t_{j+1} s'_{j+2} \ldots s'_m \rangle$ be two paths in G_F that differ only in the omission of transition t_j. Path σ and σ' yield equivalent diagnoses if $\langle s'_m, t_j, s_m \rangle$ is in G_F. Furthermore, if $s_m = s'_m$, then $\sigma \preccurlyeq \sigma'$.*

Proof. Paths σ and σ' differ only in the transition at position j and possibly the subsequent states. Since applying t_j in s'_m yields the same state as σ, i.e., s_m, the transitions $t_k, k > j$, are independent of t_j and hence the transitions commute. This independence induces equivalence classes of minimal diagnoses where each class includes diagnoses that differ only in the trace position where t_j is assumed. The second part of the theorem follows from $\epsilon \preccurlyeq t_j$.

Paths are further pruned by utilizing symmetries in the system model that arise when multiple instances of the same component class coexist in the system.

Theorem 3. *Let C^i and C^j, $C^i = C^j$, be two identical copies of the same LTS in the system model, and let δ be a diagnosis that contains transitions $t^i_{\{1\ldots m_i\}}$ and $t^j_{\{1\ldots m_j\}}$ associated with C^i and C^j, respectively. Then δ' obtained from δ by replacing $t^i_{\cdot}$ and $t^j_{\cdot}$ with the corresponding $t^j_{\cdot}$ and $t^i_{\cdot}$ is a diagnosis.*

Proof. Since $C^i = C^j$, any transition t^i enabled in a state s^i in C^i has a corresponding enabled transition t^j in a state s^j in C^j. This property transfers from individual LTSs C^i to G_F, as the i^{th} and j^{th} component of the global states cannot be modified by transitions not in C^i and C^j. It follows that a canonical representation of equivalent diagnoses (and their prefixes) can be obtained by imposing a fixed ordering on the LTS in the system model.

From the properties of $\preccurlyeq$ it follows that the minimal diagnosis prefixes can be enumerated using a best-first strategy. An incremental approach to constructing an LTS that accepts the minimal diagnoses is appropriate, since in most diagnosis scenarios a set of leading diagnoses is typically preferred to computing all possible diagnoses.

Our diagnosis algorithm starts in the global state $s_0 = \left(s_0^1, \ldots, s_0^n\right)$ and unfolds G_F only as much as is necessary to account for the next event in the observed trace r. Transitions and states are generated in best first order with respect to $\preccurlyeq$. Theorems 1–3 are applied in order to avoid generating non-minimal and equivalent solutions.

The relevant reachable part of G_F is identified starting by applying the synchronous product operation transition by transition, starting in s_0. Only paths that end with a

transition labeled with an event equivalent to the observed event e and transitions representing faults are considered. The expansion of a prefix is suspended if a successful extension path has been found. Each successful synchronized path constitutes a new extension σ' of σ that yields a state s' and a remaining observed trace suffix $r_{\delta''}$ where the initial e has been removed: $r_\delta = e.r_{\delta''}$. Once a prefix is extended, its ancestors on the path from s_0 to the expanded state must also be expanded further to maintain the frontier of minimal prefixes.

The resulting prefixes are organized in a LTS graph where event sequences leading to identical global states end in the same vertex. Hence prefixes that result in the same global state are *not* expanded multiple times. Each vertex is associated with the global state s in G_F, the remaining observed events $r_{\delta'}$ and the fault transitions in the best prefix leading to that state.

Once all events in r have been consumed a diagnosis has been found. Since the diagnosis prefixes and reachable states in G_F are expanded in best-first order, the first diagnosis is indeed a minimal diagnosis. To compute further diagnoses, earlier suspended prefix extension operations must be resumed in best-first order.

5 Evaluation

We conducted an empirical evaluation of the algorithm on a generalized version of the example given in the Fig. 1. The automaton for the threads was modified such that the lock can be re-acquired after release. Furthermore, the number of automata, number of faults in the trace, and the length of the trace was varied to measure the algorithm's performance (in terms of CPU time) and the number of minimal diagnoses. We tested different numbers of locks, critical sections and inconsistent data accesses. The results obtained from our implementation in Prolog are shown in the table below. The columns show (from left to right) the number of automata, faults, trace length, the number of minimal diagnoses and CPU time in seconds. For simplicity our implementation relies on an iterated depth-first strategy rather than pure best first search. The results were obtained on SWI Prolog 5.7.11 on an Intel Core2 CPU @1.8Ghz running Linux 2.6.30.

The results show that increasing the number of automata (left table) and trace length (top rows in the right table) has little impact on the algorithm. The number of explanations remains small and the result is available within a fraction of a second. Increasing the number of faults (bottom rows in the right table) however dramatically increases the

Automata	Faults	Length	Diags	Time (s)
4	2	20	4	0.05
12	2	20	4	0.12
52	2	20	4	0.42
5	2	20	6	0.08
7	2	28	8	0.17
9	2	36	10	0.27
11	2	44	12	0.41

Automata	Faults	Length	Diags	Time (s)
3	2	20	4	0.04
3	2	40	4	0.09
3	2	60	4	0.13
4	3	16	32	0.21
5	4	22	151	0.89
6	5	28	732	4.87

Fig. 2. Results for the "Threads and Locks" benchmark problem

computation time and number of diagnoses. This can be explained by the pathological example, where each individual fault can be explained by a number of non-equivalent faults. The results confirm that the approach is suitable for the diagnosis of typical event traces stemming from the execution of distributed loosely coupled systems, such as web services.

6 Related Work

The work by Soldani [12] is most closely aligned with ours in that the framework also relies on automata specifications and models of fault classes to identify whether a system operates normally or if a fault has occurred. Our approach generalizes this work from a single fault event to multiple missing or extraneous events. Furthermore, our approach provides more detailed fault explanations and constructs the minimal diagnoses incrementally instead of relying on a diagnoser automaton built off-line.

Yilmaz and Williams [14] employ automata models to identify possible errors in software component implementations. A parametric finite state machine model is mutated to reflect possible faults, which are subsequently confirmed or refuted by comparing the mutated model to an execution trace obtained from the implementation. Our approach employs a weaker system model but allows for multiple LTSs and accounts for limited observability in observed events.

Discrete event systems have been a common tool to monitor executions of systems [4,13] and diagnose possible faults [10]. Different from monitoring, we aim to fully explain every observed event rather than recognise known patterns. Our approach adopts a more flexible fault model than earlier work in that no a-priori limitation on possible fault transitions is necessary. Compared to dependency-based fault isolation our approach provides more detailed explanations at the sub-component level.

Similarity-based debugging of programs aims to infer possible faults in programs from observed correct and incorrect execution traces [1]. Our work differs in that faults are explained by event sequences and not simple likelihood estimates associated with individual program elements.

Sequence mining [7] has also been proposed to infer possible event sequences that are likely to lead to an error from a set of execution traces. We in contrast deal with multiple faults in a single event trace. (The approach can be extended to multiple traces.)

7 Conclusion

We introduced a diagnosis model for discrete event systems that can infer diagnoses and possible repairs in abstract event traces. The model builds on a suite of synchronized transition systems which together determine the normal as well as possible abnormal system behaviors. Our fault models are generic and are phrased in terms of added and removed transitions in the observed trace. We showed that possible explanations can be inferred locally without building the entire system model, guided by the events observed in the trace. Our algorithm interleaves the association of events to a system component, the synchronization of component models, and the resolution of discrepancies. Only distinguished explanations are computed to focus on classes of potential repairs.

In this work leading diagnoses are identified based on the assumption that faults are independent. This assumption may be relaxed in future work in order to account for systematic faults in an implementation that occur in every execution context. Furthermore, combining the event-based technique with other probabilistic debugging approaches is an avenue for further research.

References

1. Abreu, R., Zoeteweij, P., van Gemund, A.J.C.: A new bayesian approach to multiple intermittent fault diagnosis. In: IJCAI, pp. 653–658 (2009)
2. Baah, G., Podgurski, A., Harrold, M.: The probabilistic program dependence graph and its application to fault diagnosis. IEEE TSE (2010)
3. Bessey, A., Block, K., Chelf, B., Chou, A., Fulton, B., Hallem, S., Henri-Gros, C., Kamsky, A., McPeak, S., Engler, D.: A few billion lines of code later: using static analysis to find bugs in the real world. Commun. ACM 53(2), 66–75 (2010)
4. Cauvin, S., Cordier, M.O., Dousson, C., Laborie, P., Levy, F., Montmain, J., Porcheron, M., Servet, I., Trave-Massuyes, L.: Monitoring and alarm interpretation in industrial environments. AI Communications 11(3/4) (1998)
5. Chen, C.H., Yücesan, E.: Introduction to discrete event systems: Christos G. cassandras and stephane lafortune. Kluwer Academic Publishers, Dordrecht (1999); Automatica 37(10), 1682–1685 (2001)
6. Friedrich, G., Fugini, M.G., Mussi, E., Pernici, B., Tagni, G.: Exception handling for repair in service-based processes. IEEE TSE (2010)
7. Hsu, H.Y., Jones, J.A., Orso, A.: Rapid: Identifying bug signatures to support debugging activities. In: ASE, pp. 439–442. IEEE, Los Alamitos (2008)
8. Liu, Y.: A formalization of program debugging in the situation calculus. In: Fox, D., Gomes, C.P. (eds.) AAAI, pp. 486–491. AAAI Press, Menlo Park (2008)
9. Mayer, W., Stumptner, M., Wotawa, F.: Can AI help to improve debugging substantially? automatic debugging and the jade project. Journal of the Austrian Society for Artificial Intelligence 21(4), 18–22 (2002)
10. Pencolé, Y., Cordier, M.O.: A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks. Artif. Intell. 164(1-2), 121–170 (2005)
11. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence 32, 57–95 (1987)
12. Soldani, S., Combacau, M., Subias, A., Thomas, J.: Intermittent fault diagnosis: a diagnoser derived from the normal behavior. In: Proc. DX 2007 (2007)
13. Yan, Y., Dague, P., Pencolé, Y., Cordier, M.O.: A model-based approach for diagnosing fault in web service processes. Int. J. Web Service Res. 6(1), 87–110 (2009)
14. Yilmaz, C., Williams, C.: An automated model-based debugging approach. In: ASE, pp. 174–183. ACM Press, New York (2007)

Evaluating Sequential Single-Item Auctions for Dynamic Task Allocation

Adrian Schoenig and Maurice Pagnucco

National ICT Australia and ARC Centre of Excellence in Autonomous Systems
School of Computer Science and Engineering, UNSW
Sydney, NSW, 2052, Australia
{adrians,morri}@cse.unsw.edu.au

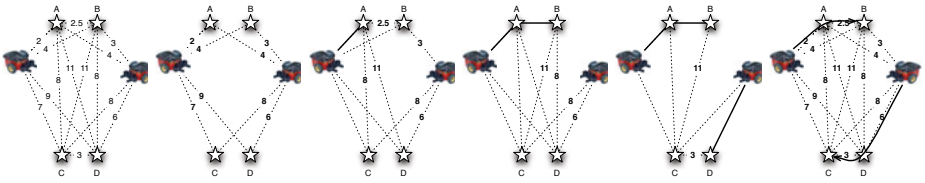
Abstract. *Sequential single-item* (SSI) auctions have proven a very effective technique for tackling *static* task allocation problems in multi-robot settings and are only recently being applied to *dynamic task allocation problems*. We complement existing work by evaluating the effects of using different auctioning and winner determination schemes when dealing with dynamically appearing tasks. To this end we investigate the use of plan modification versus re-planning and minimum cost and regret clearing for winner determination in the auction.

1 Introduction

The wisdom of distributing a task is strongly established in the well-known English proverbs "two heads are better than one" and "many hands make light work". However, the mere weight of numbers is not in itself sufficient to complete tasks in an optimal or even efficient manner. To this end numerous *co-ordination* techniques have been proposed for multi-robot systems so that what can be achieved by the robots in combination is more than just the sum of their individual actions. Technically, co-ordination can be viewed as a distributed goal search problem. Given a common goal that can be decomposed into sub-tasks and several team members, each of which may have differing resources and abilities, determine the allocation of sub-tasks to each team member so that the common consumption of resources is optimised.

Recently, market-based approaches have become a popular means of achieving co-ordination [1]. Here we examine the *sequential single-item* (SSI) auction scheme which has been successfully applied to co-ordinating multiple robots where tasks are known at the outset. One of the main advantages of market-based approaches is that they can incorporate new tasks easily. However, the effects of this on the optimality of the resulting assignment have not been studied yet in detail. *The contribution of this paper is an investigation of the efficiency of different approaches for applying sequential single-item auctions in* dynamic *task allocation problems where not all tasks are know at the start of the auction. In particular, we investigate the use of plan modification versus re-planning and minimum cost and regret clearing for winner determination in the auction. We*

J. Li (Ed.): AI 2010, LNAI 6464, pp. 506–515, 2010.



(a) Setting (b) Round 1 (c) Round 2 (d) Round 3 (e) Round 4 (f) Result

Fig. 1. Example setting with rounds and result of a sequential auction. Dotted edges are costs, full edges are assigned paths.

present empirical results that demonstrate that SSI auctions can indeed be applied effectively to dynamic task allocation and under some circumstances even result in improvements compared to static allocation.

2 Multi-robot Co-ordination

Multi-robot co-ordination is often achieved by organising robots within a structural framework. The most common frameworks are motivated by social structures studied in psychology, economics and sociology and, not surprisingly, widely adopted in team sports. Parker [2] provides a survey which also includes bio-inspired approaches such as swarms and knowledge sharing approaches.

Auctions are a common market-based methodology by which robots can share tasks. Extending the economic analogy, robots are traders, tasks are commodities and each task has a cost (as well as, possibly, a reward). Robots buy and sell tasks in an attempt to optimise their individual profits, which in total leads to a maximisation of the overall team profit. At the cost of a high communication overhead, market-based approaches enjoy the advantages that robots can trade tasks among one another, co-ordination can be fully decentralised, heterogeneous systems are easily supported, and dynamic events can be responded to quickly.

A typical auction is composed of three phases: In the *initial phase* the auctioneer sends a request to all robots notifying them of the tasks that are for sale. This is followed by the *bidding phase*, during which the robots evaluate the tasks, calculate bids for tasks in which they are interested, and return them to the auctioneer. Finally, during the *winner determination phase* the auctioneer determines the winner for each of the tasks and notifies the winning robots.

Several protocols have been investigated for conducting auctions where multiple tasks and multiple robots are at the auctioneer's disposal. They can be categorised into three classes: *parallel*, *combinatorial* and *sequential* auctions. Sequential auctions are the market-based scheme we adopt so we will consider them in more detail below. Briefly however, in parallel auctions each robot calculates one bid for every task and the auctioneer assigns all tasks at once. The example in Figure 1 consists of 4 tasks A, B, C, and D, and uses the MiniSum team objective where the aim is to minimise the sum of costs for completing all tasks. Using travel distance as the cost measure, tasks A and C are assigned to

the left robot (for a cost of $2+7=9$) and tasks B and D to the right robot (for a cost of $6+3=9$) giving a team cost of 18. The computational complexity of this protocol is $O(\#robots*\#tasks)$ but the solutions are likely to be sub-optimal. In combinatorial auctions each robot calculates a bid for every subset of the tasks on offer. In our example, the task set $\{A, B\}$ is assigned to the left robot (for a cost of $2+2.5=4.5$) and $\{C, D\}$ to the right robot (for a cost of $3+6=9$) giving a total team cost of 13.5. In this scheme 2^n-1 bids are required when n tasks are on offer so the computational complexity is a prohibitive $O(2^{\#tasks} * \#robots)$.

Sequential auctions are in the middle ground between parallel and combinatorial auctions: the auction proceeds over several rounds and a subset of tasks (typically one) is assigned to a robot in each round. At the start of each round, all remaining tasks are advertised to all robots. Each robot returns a bid to complete a single task in addition to those it has committed to in previous rounds. During the winner determination phase the auctioneer selects one task and assigns it to one robot (e.g., the lowest bidder). This particular type of auction is referred to as a *sequential single-item* (SSI) auction. It starts with a partial solution to the problem which is then iteratively extended to a complete solution. Figure 1 presents an example. After the first round task A is assigned to the left robot as for the parallel auction. In the second round, tasks B, C, and D are on offer. The left robot bids 2.5 to complete task B (cost reduced since it has already been assigned task A) and the right robot bids 3 for task B. The auctioneer therefore assigns task B to the left robot. In the last two rounds task C (with cost 6) and then task D (with additional cost 3) are assigned to the right robot. The total team cost is 13.5, which is the same as the optimal solution obtained by the combinatorial auction. However, sequential auctions are in general not guaranteed to find the optimal solution. Instead, they provide clearly better solutions than parallel auctions as synergies between tasks can be exploited. This comes at a computation cost of $O(\#tasks^2*\#robots)$, which is a significant improvement over combinatorial auctions.

We consider SSI auctions here in a similar manner to [3,4,5] and also investigate *regret clearing* [6], which is an alternative winner determination mechanism in which the auctioneer (and hence the team) prefers assignments that will be regretted the least, i.e., those unlikely to change during the auction. The auctioneer assigns the task that maximises the difference between the lowest and the second lowest bid rather than the one minimising the team objective, as this was shown to improve over standard minimum cost winner determination.

Auctions should be designed that they minimise a global team objective. One possibility is to minimise the sum of all costs of all robots, which is called the *MiniSum* team objective. In this case, robots bid the increase in their cost under the assumption that the task(s) will be awarded to them. Alternatively, the *MiniMax*/*MiniAve* objectives minimise the maximum/average cost over all robots in order to minimise the total/average completion cost.

Note: we have not considered rewards for achieving goals. We do so purely to consider whether SSI auctions can prove effective in the simpler setting without rewards, rather than complicating matters unnecessarily and leave the more complex case to future work.

3 Dynamic Task Allocation

As outlined in a recent survey [1], dynamic task allocation has been investigated in a number of works. However, our approach differs in that it is based on sequential single-item auctions, which have been shown to provide a good compromise between computational complexity and solution quality.

The question we pose here is *whether SSI auctions can be effectively applied to dynamic task allocation problems where tasks to be completed may not all be known at the start of the auction* (i.e., so-called *online* tasks)

Similar to our approach, is the work by Nanjanath and Gini [7,8] who investigate dynamic task allocation using sequential auctions. Their focus lies on addressing robustness issues caused by unreliable robots using sequential auctions, whereas our focus is on the optimality with respect to dynamically appearing tasks. Our work can be seen as complementary as we provide an evaluation of different auction schemes in the presence of online tasks.

There are several considerations distinct to task allocation of online tasks that do not arise in the static case:

- Since new tasks can appear at any time it is difficult to define a short-term team objective. Task allocation should seek to minimise the overall time taken to complete all tasks and, as a result, the MiniSum heuristic is suboptimal as it can lead to tasks being completed by only few robots with many robots being idle. MiniMax and MiniAve can prevent this situation.
- The location where a robot ends up after completing its assigned tasks is important when subsequent tasks appear. A more central position is likely to provide better performance.
- Therefore robots should re-order yet to be completed tasks as new tasks are assigned to them. We adopt a variant of the *cheapest insertion heuristic*[1] that allows re-ordering the robot's previously last task. Thus the robot can re-evaluate which task to fulfil last as this decision has high influence on its cost for reaching tasks that appear later.

Furthermore, it is not clear in dynamic task allocation whether it pays to incrementally adjust the tasks assigned to each robot or whether to put all tasks up for re-consideration. In the context of planning, Nebel and Koehler [9] have proved that plan repair and re-planning belong to the same complexity class in the worst case. Accordingly, we implemented and evaluated the following two dynamic task allocation schemes:

Re-planning. When a new task arrives, all previously assigned but uncompleted tasks are put up for auction once again.[2] This results in the same allocation as auctioning all available tasks at once as in [4,5,6,7,8].

[1] The cheapest insertion heuristic ensures that robots maintain their current task ordering and insert new tasks into this ordering at the optimal point.

[2] Re-planning also allows the robots to re-compute all task costs. However, this is not relevant in our setting as we do not simulate robot movement as described later.

Table 1. Comparison of using different heuristics for bid calculation using MiniMax team objective. Selling a total of $n = 10$ tasks to $m = 3$ robots.

Setting		Minimum cost				Regret clearing			
Heuristic	k	Time	ΔTime	Cost	ΔCost	Time	ΔTime	Cost	ΔCost
All permutation	10	1.78s	-	67.1	-	1.03s	-	58.8	-
	5	1.53s	-	64.3	-	1.36s	-	59.9	-
	3	1.12s	-	61.2	-	1.13s	-	60.2	-
	2	1.22s	-	61.7	-	1.03s	-	59.7	-
	1	1.09s	-	61.8	-	1.11s	-	61.8	-
Modified cheapest insertion	10	1.05s	-40.9%	67.1	0.00%	1.03s	-36.4%	59.4	0.92%
	5	0.91s	-40.2%	64.3	0.00%	0.91s	-32.7%	59.9	0.00%
	3	0.78s	-30.3%	61.2	0.00%	0.77s	-31.9%	60.2	0.00%
	2	0.74s	-39.5%	62.0	0.47%	0.72s	-30.1%	59.7	0.00%
	1	0.64s	-40.1%	61.8	0.00%	0.65s	-41.5%	61.8	0.00%
Standard cheapest insertion	10	0.62s	-65.1%	68.4	1.67%	0.63s	-60.8%	60.6	2.98%
	5	0.67s	-56.3%	64.8	0.79%	0.67s	-50.7%	60.6	0.55%
	3	0.63s	-43.9%	61.3	0.07%	0.62s	-45.0%	60.2	0.00%
	2	0.59s	-51.4%	62.5	1.36%	0.60s	-42.2%	59.9	0.20%
	1	0.60s	-44.6%	62.0	0.28%	0.60s	-45.5%	62.0	0.28%

Plan Modification. Robots retain their previously assigned tasks and bid for the additional cost of undertaking the newly generated task(s).

The focus of this paper is on evaluating how these different dynamic task allocations schemes perform in the setting of online tasks in SSI auctions. While SSI auctions allow by design to easily add new tasks dynamically, we investigate the effect that this has on the optimality of the solutions and how they compare to static assignment with all tasks known in advance. In particular, we evaluate how plan modification compares to re-planning, as this is a direct comparison between dynamic and static task allocation. Furthermore, we compare minimum cost and regret clearing winner determination in both settings.

4 Experimental Setting

In order to focus on the research question at hand we make these assumptions:

1. Tasks can be achieved by a single robot and are atomic, i.e., do not have to be further decomposed.
2. Tasks are loosely coupled (no tight co-ordination).
3. All robots bid the increase in their costs for reaching the advertised tasks.
4. Robots are provided with a map and have perfect localisation.
5. Robots do not wear out.
6. There is no centralised server.
7. New tasks can appear dynamically at any time.

Assumptions 1–5 are standard assumptions made by previous work on SSI auctions, e.g., [4,5,6,7,8]. Nanjanath and Gini [7,8] also address Assumption 6 and additionally deal with unreliable execution by the robots. Assumption 7 is new to this work.

We focus on online tasks, i.e., dynamically new appearing tasks and evaluate the effect this has on the resulting assignments from the SSI auctions. The question we investigate is how much worse SSI auctions fare, if not all tasks can be bid for at once as they are not known from the start of the auctions and therefore sets of tasks have to be auctioned in subsequent rounds. As a simulation of the dynamic scenario, we split the existing tasks into a number of subsets, and sell all tasks of one subset before continuing with the next subset until every task has been assigned to a robot.

The physical movement of the robots can have significant effects on the run time of the experiments that is likely to introduce random unexpected situations such as robots getting stuck, which makes it prohibitive to rerun and compare same experiments with varying auction schemes. As we are mainly interested in the effect online tasks have on the optimality of the SSI auction, we do not simulate the movement of the robots and thus the execution of the tasks. A good discussion of addressing the robustness of sequential auctions in such cases can be found in [7,8].

The setting used for our experiments is a realistic floor plan of a hospital that comes with the Player/Stage [10] robot middle-ware. 26 distinct positions are identified on the map. The maps used by Koenig *et al.* [4] are simpler with fewer positions although they have the ability to modify their maps by randomly opening and locking 'doors' on the map.

Experiments were conducted for combinations of m robots $R = \{r_1, \ldots, r_m\}$ and n tasks $T = \{t_1, \ldots, t_n\}$ assigning k tasks $K \subseteq T$ in each round, with $m \in \{3, 5, 10\}$, $n \in \{10, 15, 20\}$, and $k \in \{1, 2, 3, 5, n\}$. Each setting was evaluated using 25 randomly generated scenarios (as in Koenig [6]) and results averaged. Moreover, the scenarios are "standardised" by incrementally adding robot positions and task positions as these are increased in the experiments noted previously[3], i.e., if $|R_1| < |R_2|$ then $R_1 \subset R_2$, and equally for T_1 and T_2. Every robot can initiate new auctions and act as a participant. A centralised auctioneer is therefore not required. In our simulation, we let an arbitrary robot know of new tasks, which then starts a new round of auctions.

We began by first reproducing results from previous work on SSI auctions by Koenig and Zheng and their collaborators [3,4,6]. Our experiments showed the same trends, most notably that regret clearing provided a clear advantage over minimum cost winner determination for the MiniMax team objective. In order to determine the effectiveness of SSI auctions for dynamic task allocation we ran experiments using both the *minimum cost* and the *regret clearing* winner determination methods as well as the *re-planning* and *plan modification* dynamic task allocation schemes.

[3] This allows us to directly compare the resulting assignments from experiments with different number of robots and tasks.

5 Experimental Results and Evaluation

Our first experiment tests the claim that performance can be improved by modifying the *cheapest insertion heuristic* to allow re-ordering of the last task the robot intends to complete. Table 1 shows a comparison of the standard cheapest insertion heuristic against our variant. Experiments were run on 3 robots and 10 online tasks arriving dynamically in sets of size $k \in \{1, 2, 3, 5, 10\}$. The first group considers all permutations of all uncompleted and newly arrived tasks, and is thus guaranteed to find the optimal ordering. The second group represents our modified heuristic that allows the last of all committed tasks to be re-ordered, while the last group uses the standard cheapest insertion heuristic. The baseline is the first group "all permutations" as this yields the optimal outcome.

We see that our modified cheapest insertion heuristic showed no variation from the optimal in 8 out of 10 cases while the standard cheapest insertion heuristic showed variations in most cases. This leads us to conclude that our modification to the heuristic provides a significant cost improvement at only small increase in run-time. The null hypothesis that this is false can be rejected with a one-sided paired t-test with confidence 0.999 (0.995) and mean difference of 0.75% (0.38%) for minimum cost (regret clearing) winner determination .

Table 3 shows our main results for applying SSI auctions to dynamic task allocation. Comparing minimum cost with regret clearing for winner determination, the minimal values are highlighted in bold font. Underlined values indicate where dynamic task allocation resulted in an improvement over allocating all tasks at once as in the static problem setting. Table 2 compares the plan modification and re-planning task allocation schemes.

Table 2. Comparison of using plan modification and re-planning to dynamically assign $n = 15$ tasks using minimum cost winner determination. Cheapest insertion heuristic did not allow re-ordering any committed tasks.

Setting		Plan modification			Re-planning		
m	k	Time	Sum	Max	Time	Sum	Max
3	15	2.37s	208.7	81.9	2.36s	208.7	81.9
	5	1.52s	206.1	79.6	3.80s	208.7	81.9
	3	1.27s	213.4	81.9	5.30s	208.7	81.9
	2	0.97s	217.3	84.0	7.50s	208.7	81.9
	1	0.86s	218.9	82.2	11.76s	208.7	81.9
5	15	2.84s	189.3	52.9	2.82s	189.3	52.9
	5	1.44s	190.2	51.1	3.85s	189.3	52.9
	3	1.23s	195.1	51.0	5.75s	189.3	52.9
	2	1.05s	199.9	53.3	8.78s	189.3	52.9
	1	0.92s	203.2	54.2	14.55s	189.3	52.9
10	15	3.79s	128.8	30.0	3.73s	128.8	30.0
	5	2.14s	140.1	28.1	6.04s	128.8	30.0
	3	1.72s	144.2	28.1	8.55s	128.8	30.0
	2	1.50s	148.6	28.5	14.54s	128.8	30.0
	1	1.13s	149.1	28.3	21.05s	128.8	30.0

Table 3. Experimental results of using plan modification for dynamic task allocation using MiniMax team objective. Cheapest insertion heuristic allowed re-ordering the last committed task.

Setting			Minimum cost		Regret clearing		M.C. versus R.C.
n tasks	m robots	k tasks/round	Cost	ΔCost	Cost	ΔCost	ΔCost
10	3	10	67.1	-	**60.1**	-	-10.4%
		5	64.3	-4.2%	**59.9**	-0.4%	-6.9%
		3	61.2	-8.8%	**60.2**	0.1%	-1.7%
		2	62.0	-7.7%	**59.7**	-0.7%	-3.7%
		1	61.8	-7.9%	61.8	2.8%	0.0%
	5	10	45.0	-	**40.7**	-	-9.5%
		5	43.5	-3.4%	**40.7**	0.0%	-6.3%
		3	42.5	-5.6%	**42.0**	3.2%	-1.1%
		2	44.0	-2.2%	**42.4**	4.1%	-3.6%
		1	42.9	-4.7%	42.9	5.3%	0.0%
	10	10	26.7	-	**24.3**	-	-9.1%
		5	26.3	-1.3%	**25.3**	4.5%	-3.8%
		3	26.0	-2.7%	**25.4**	4.9%	-2.0%
		2	26.4	-1.2%	**25.8**	6.4%	-2.1%
		1	26.1	-2.1%	26.1	7.6%	0.0%
15	3	15	80.8	-	**76.0**	-	-6.0%
		5	83.2	3.0%	**79.0**	4.0%	-5.1%
		3	79.2	-1.9%	**77.9**	2.5%	-1.7%
		2	82.6	2.3%	**79.5**	4.7%	-3.8%
		1	82.3	1.9%	82.3	8.4%	0.0%
	5	15	54.4	-	**50.4**	-	-7.4%
		5	53.3	-2.1%	**51.0**	1.1%	-4.4%
		3	55.3	1.6%	**52.6**	4.2%	-4.9%
		2	56.4	3.7%	**55.4**	10.0%	-1.7%
		1	55.7	2.3%	55.7	10.5%	0.0%
	10	15	30.5	-	**28.3**	-	-7.0%
		5	31.5	3.3%	**30.4**	7.3%	-3.4%
		3	30.6	0.5%	**30.3**	6.9%	-1.2%
		2	30.7	0.8%	**30.3**	6.9%	-1.4%
		1	30.4	-0.1%	30.4	7.4%	0.0%
20	3	20	98.8	-	**89.9**	-	-9.0%
		5	98.1	-0.7%	**92.6**	2.9%	-5.7%
		3	93.1	-5.8%	**91.2**	2.3%	-1.1%
		2	95.1	-3.7%	**94.1**	4.6%	-1.1%
		1	97.0	-1.8%	97.0	7.8%	0.0%
	5	20	62.5	-	**58.9**	-	-5.6%
		5	63.0	0.9%	**60.2**	2.1%	-4.5%
		3	61.2	-2.1%	61.2	3.8%	0.0%
		2	63.5	1.7%	**63.0**	7.0%	-0.8%
		1	63.0	0.9%	63.0	7.0%	0.0%
	10	20	34.2	-	**31.3**	-	-8.7%
		5	34.9	2.0%	**33.4**	6.7%	-4.4%
		3	33.1	-3.3%	33.1	6.0%	0.0%
		2	33.4	-2.6%	**33.0**	5.6%	-1.0%
		1	33.2	-3.1%	33.2	6.1%	0.0%

The following results are particularly noteworthy:

Minimum Cost and Plan Modification. SSI auctions using minimum cost winner determination and plan modification outperform re-planning in the majority of cases. This is due to SSI auctions not being guaranteed to find the optimal solution since not all synergies between tasks can be considered by the robots in the auction. Assignments based on SSI auctions can get stuck in local minima instead of converging to the global minimum. The higher the ratio of tasks to robots, the more likely SSI auctions are to result in sub-optimal solutions. When randomly drawn subsets of tasks are up for bid however, as in the dynamic setting, these local minima can be avoided to some extent. However, the hypothesis that dynamic task allocation using plan modification results in an improvement has been statistically rejected when tested using a one-sided t-test and a 5% significance level. So is the hypothesis that it is worse than static assignment. Our experiments showed therefore that dynamic task allocation is statistically no worse than static task allocation when using minimum cost winner determination, and can even result in an improvement.

Regret Clearing and Plan Modification. Assigning tasks dynamically using regret clearing winner determination and plan modification provides worse results both in terms of the maximums and the sums than using re-planning. In contrast to minimum cost winner determination, regret clearing rarely produced results where plan modification performed better than re-planning. The argument can be made that this follows from the reason given by Koenig *et al.* [6] for using regret clearing in the first place. Its design goal was to assign the tasks in such a way that those assignments which would be least regretted are made first. This encourages avoiding local minima in a similar way to randomly drawn subsets for dynamic allocation. The additional random partitioning that is caused by dynamic task allocation with plan modification thus only marginally improves the assignment.

Minimum Cost vs Regret Clearing. Regret clearing performed strictly better in most instances. We evaluated the hypothesis that regret clearing results in lower team costs for the MiniMax team objective rather than minimum cost winner determination using a one-sided paired t-test. The null hypothesis that this is not the case can be rejected with confidence 0.999 and a mean difference of 3.41%. Based on our experiments, regret clearing is the preferred method for dynamic task allocation, though the improvements compared to minimum cost determination are not as significant as when applied to static task allocation.

6 Conclusions and Further Work

We have investigated SSI auctions applied to dynamic task allocation with online tasks and demonstrated that they indeed provided a powerful framework in this setting. In our empirical trials, we compared plan modification

to re-planning and minimum cost winner determination to regret clearing. Plan modification provided significant improvements in run-time, while resulting in reasonable assignments. A surprising insight was that it could even result in improvements in the resulting team costs if the ratio of tasks per robot is relatively high and when used in combination with minimum cost winner determination. We conclude that this is caused by the fact that random subsets of tasks can help to avoid local minima, in which the hill-climbing nature of sequential auctions is destined to get trapped. However, regret clearing provided consistently better assignments than minimum cost winner determination while having similar running times.

For future work we think the following improvements regarding bidding and selling are promising. For bidding, *clustering tasks* based on their proximity makes sense as this allows capturing the "synergies" between tasks. This additional information could help rule out combinations which are unlikely to win. For selling, giving robots the option to *resell tasks* which they previously committed to could improve the performance as new tasks are dynamically allocated. This builds on related work [7,8] and requires an efficient way of determining which tasks should be re-sold to avoid starting too many auctions.

References

1. Dias, M.B., Zlot, R., Kalra, N., Stentz, A.: Market-based multirobot coordination: A survey and analysis. Proc. IEEE 94(7), 1257–1270 (2006)
2. Parker, L.: Distributed intelligence: Overview of the field and its application in multi-robot systems. J. Physical Agents 2(1), 5–14 (2008)
3. Lagoudakis, M., Markakis, E., Kempe, D., Keskinocak, P., Kleywegt, A., Koenig, S., Tovey, C., Meyerson, A., Jain, S.: Auction-based multi-robot routing. In: Proc. Int. Conf. on Robotics: Science and Systems, pp. 343–350 (2005)
4. Koenig, S., Tovey, C., Lagoudakis, M., Markakis, V., Kempe, D., Keskinocak, P., Kleywegt, A., Meyerson, A., Jain, S.: The power of sequential single-item auctions for agent coordination. In: Proc. AAAI 2006, pp. 1625–1629 (2006)
5. Zheng, X., Koenig, S., Tovey, C.: Improving sequential single-item auctions. In: Proc. IROS 2006, pp. 2238–2244 (2006)
6. Koenig, S., Zheng, X., Tovey, C., Borie, R., Kilby, P., Markakis, V., Keskinocak, P.: Agent coordination with regret clearing. In: Proc. AAAI 2008 (2008)
7. Nanjanath, M., Gini, M.: Auctions for task allocation to robots. In: Proc. Int. Conf. on Intelligent Autonomous Systems (January 2006)
8. Nanjanath, M., Gini, M.: Performance evaluation of repeated auctions for robust task execution. In: Carpin, S., Noda, I., Pagello, E., Reggiani, M., von Stryk, O. (eds.) SIMPAR 2008. LNCS (LNAI), vol. 5325, pp. 317–327. Springer, Heidelberg (2008)
9. Nebel, B., Koehler, J.: Plan modification versus plan generation: A complexity-theoretic perspective. In: IJCAI 1993, pp. 1436–1441 (1993)
10. Gerkey, B., Vaughan, R., Howard, A.: The player/stage project: Tools for multi-robot and distributed sensor systems. In: Proc. ICRA 2003 (January 2003)

Maximal Matching for Double Auction*

Dengji Zhao[1,2], Dongmo Zhang[1], Md Khan[1], and Laurent Perrussel[2]

[1] Intelligent Systems Laboratory,
University of Western Sydney, Australia
{dzhao,dongmo,md.khan}@scm.uws.edu.au
[2] IRIT, University of Toulouse, France
laurent.perrussel@univ-tlse1.fr

Abstract. We study the problem of mechanism design for a double auction market where multiple buyers and sellers buy and sell a commodity. We design and implement a matching algorithm that maximizes market liquidity, including the number of transactions and buy/sell-volume. We prove that, given the number of matches, the algorithm also maximizes auctioneer's profit. Based on the CAT Tournament (Trading Agent Competition Market Design) platform, we show with experiments that the new matching method not only increases market liquidity but also significantly improves market share and auctioneer's profit in the long term, compared with equilibrium matching, the most commonly used matching method.

1 Introduction

A double auction is a market mechanism that allows multiple buyers and sellers trade simultaneously [1,2,3]. Given the supply and demand of sellers and buyers (i.e. traders), a double auction is characterized by (i) how to match bids (offers to buy) and asks (offers to sell), and (ii) what price to execute each matched ask-bid pair (clearing price). Similar to the design of other market mechanisms, the main concerns of double auction design include *incentive compatibility* (IC), *market liquidity*, *market efficiency* (social welfare) and *market profit* (auctioneer's revenue). Although all these properties are desirable, it is impossible for a double auction mechanism to possess all of them. McAfee and Wurman *et al.* showed that there is no double auction mechanism that is both efficient and incentive compatible [4,5].

As a tradition in the research of mechanism design, most existing work on double auctions put emphasis on incentive compatibility and social welfare (e.g. [4,5,3]). However, the other properties, especially market liquidity, are equally important. The liquidity of a market, which indicates the number of transactions and buy/sell volume of the market, not only effects market profit and social welfare but also flags the success and reputation of the market. A market with high liquidity can attract traders to the market as it brings trading opportunities.

* This research was supported by the Australian Research Council through Discovery Project DP0988750.

J. Li (Ed.): AI 2010, LNAI 6464, pp. 516–525, 2010.

This paper makes an endeavour at the design of non-IC double auction mechanism. Without restriction of incentive compatibility, we can maximize matches and set clearing prices in terms of bidding prices, therefore it becomes possible to maximise market liquidity and optimize other market indicators, such as market share, market profit and social welfare. To this end, we propose a new matching algorithm, named *maximal matching* (MM), for any double auction mechanism and prove that it maximises the matches of incoming bids and asks. By comparing the algorithm with the most commonly used matching method, *equilibrium matching* (EM), we demonstrate through a set of experiments that maximal matching not only maximizes market liquidity, but also significantly improves market share (in terms of the number of traders) and auctioneer's profit.

The rest of the paper is organized as follows. Section 2 introduces the market model and lists the commonly used criteria for double auction mechanism design. Section 3 presents maximal matching algorithm. Section 4 analyses the properties of maximal matching and compares maximal matching with equilibrium matching. Section 5 shows the experimental results of both maximal and equilibrium matching, which gives a support to the theoretical results presented in Sect. 4. Finally we conclude the work.

2 Preliminaries

2.1 The Market Model

We study the problem of matching in a double auction market where multiple buyers and sellers buy and sell one commodity. Let $T = S \cup B$ be a set of traders, where S is the set of sellers, B is the set of buyers, and $S \cap B = \varnothing$[1]. A *shout* is a message that a trader sends to the auctioneer (the double auction market) for either buying or selling one unit of the commodity with a specified price[2]. Let Ω be the set of all possible shouts. For each shout $s \in \Omega$, we write $p(s)$ to represent the bidding price of s (non-negative), and $t(s)$ to denote the trader who sends s.

A double auction is running as follows: each trader submits a number of shouts to the auctioneer, and the auctioneer decides which sell and buy shouts to be matched and what price to execute each match.

Definition 1. *An **ask** is a shout $a \in \Omega$ such that $t(a) \in S$. A **bid** is a shout $b \in \Omega$ such that $t(b) \in B$. For any finite set $X \subset \Omega$, we let $X^{ask} = \{x \in X : t(x) \in S\}$ and $X^{bid} = \{x \in X : t(x) \in B\}$. A **matching** of X, denoted by $M(X)$, is a collection of pairs $\{(a_1, b_1), (a_2, b_2), ...\}$, where $a_i \in X^{ask}$, $b_i \in X^{bid}$, $p(a_i) \leq p(b_i)$, and $a_i \neq a_j$, $b_i \neq b_j$ when $i \neq j$.*

[1] In the real word, a trader can be both a seller and a buyer for the same commodity. In such a case, we model it as two different roles as the decision making for selling and buying is different.

[2] For sake of simplicity, we assume that each shout contains only one unit of the commodity. It is possible to extend the model to handle a shout with multiple units by splitting the shout into many shouts with one unit each.

2.2 Objectives Considered in Double Auction Mechanism Design

The following desirable objectives or desiderata are mostly considered in double auction design [6]:

- *Incentive Compatibility.* A double auction is said to be incentive compatible if all of the participants maximize their utilities when they truthfully reveal any private information asked for by the auction.
- *Liquidity Maximization.* The goal is to maximize: (a) number of transactions, (b) sell volume: the total amount of cleared asks, and (c) buy volume: the total amount of cleared bids [3].
- *Profit Maximization.* Each pair of ask and bid that are matched produces a profit, which is the difference between the bid price and the ask price. This objective is to maximize the sum of these differences, over all matched pairs.
- *Social Welfare Maximization (Efficiency).* This objective corresponds to maximizing the goods of the buyers and sellers in aggregate. That is, the goods are allocated to the agents who value them most highly.
- *Individual Rationality.* A double auction is individual rational if it gives its traders non-negative utility/profit.

3 Matching Algorithm

This section will present our maximal matching algorithm. Before doing that, let us briefly introduce the most commonly used matching policy for double auction markets–*equilibrium matching* (EM).

3.1 Equilibrium Matching

Equilibrium Matching is used to find a *uniform price* p^* (equilibrium price) which balances the bids and the asks going to be matched so that all the bids with price $p \geq p^*$ and all the asks with price $p \leq p^*$ are matched [1]. EM can be easily implemented as follows:

1. *Sort all asks (bids) in ascending (descending) order w.r.t. their price.*
2. *Based on this sort order, starting at the top, add each ask-bid pair to the result matching, if ask's price is less than or equal to bid's price.*

Algorithm 3.1 describes the above implementation. A more deliberate, and yet popular, implementation, named *4-Heap*, is given by Wurman *et al.* [5]. A uniform price is normally determined by the last matchable or the first unmatchable shout pair w.r.t. the matching order in Algorithm 3.1 [4,5].

3.2 Maximal Matching

A double auction with EM can be incentive compatible or efficient (but not both) with some special pricing polices [4,5]. However, no double auction mechanism with EM can maximize liquidity as the uniform clearing price might prohibit some matchable shouts from being matched. In order to maximise the number

Algorithm 3.1. EquilibriumMatching

Input: *Asks*: sorted in ascending order, *Bids*: sorted in descending order
Output: *Matching*

```
1  begin
2      Matching ← ∅; I ← 1;
3      while Asks ≠ ∅ and Bids ≠ ∅ do
4          Ask ← read I-th ask from Asks; Bid ← read I-th bid from Bids;
5          if p(Ask) ≤ p(Bid) then
6              Matching ← Matching ∪ {(Ask, Bid)}; I ← I + 1;
7          else
8              jump out while loop;
9          end
10     end
11 end
```

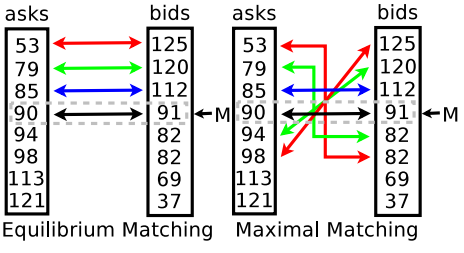


Fig. 1. Equilibrium Matching vs. Maximal Matching

of matches/transactions, it is essential to allow different matches cleared at different prices[3]. Otherwise, some matches might be cleared at a price which is not between the ask price and the bid price of them, i.e. it will act against individual rationality, which is a basic assumption of double auction mechanism design. Based on this idea, we introduce a new matching algorithm, named *Maximal Matching* (MM), as we shall prove that it maximises the number of matches in Sect. 4. The algorithm is given in Algorithm 3.2, which can be summarised in the following five steps:

1. *Given an input of shouts, calculate the matching (the set of matched pairs) with Algorithm 3.1, and mark all the matched shouts as matched and all the other shouts as unmatched (lines 4-6).*
2. *Recursively check how many matches MM can achieve if the input shouts were matched asks and unmatched bids (line 8).*
3. *Recursively check how many matches MM can achieve if the input shouts were unmatched asks and matched bids (line 9).*
4. *Choose the minimum of the numbers from the last two steps as the extra number of matches MM can achieve (line 10).*

[3] Sales of identical goods or services are transacted at different prices is named *price discrimination* [7].

5. *Cross match extra matchable shouts with the matched shouts in step 1: the ask in the first matched pair is rematched with the last extra matchable bid, while the bid in the pair is rematched with the last matchable ask, then the second matched pair with the second last extra matchable ask and bid, and so on until all extra matchable shouts are matched (lines 11-19).*

Algorithm 3.2. MaximalMatching

Input: $Asks$: sorted in ascending order, $Bids$: sorted in descending order
Output: $Matching$

```
1  begin
2      Matching ← ∅;
3      if Asks = ∅ or Bids = ∅ then Return;
4      Matching ← EquilibriumMatching(Asks, Bids);
5      MatchedAsks ← all asks from Matching in ascending order;
6      MatchedBids ← all bids from Matching in descending order;
7      if (Bids \ MatchedBids) = ∅ or (Asks \ MatchedAsks) = ∅ then Return;
8      MM1 ← MaximalMatching(MatchedAsks, (Bids \ MatchedBids));
9      MM2 ← MaximalMatching((Asks \ MatchedAsks), MatchedBids);
10     ExtraNumberOfMatches ← Min(|MM1|, |MM2|);
11     I ← ExtraNumberOfMatches; N ← |Matching| + 1;
12     while I > 0 do
13         (Ask1, Bid1) ← read I-th in match from Matching;
14         Ask2 ← read N-th ask from Asks;
15         Bid2 ← read N-th bid from Bids;
16         Matching ← Matching ∪ {(Ask1, Bid2), (Ask2, Bid1)};
17         Matching ← Matching \ {(Ask1, Bid1)};
18         N ← N + 1; I ← I − 1;
19     end
20 end
```

Figure 1 shows a matching example of both EM and MM with the same set of shouts, where the numbers are the prices of shouts (other information is omitted), M indicates the last matchable pair with EM, and the arrowed lines link each matched pair. We can see that MM achieves two extra matches than EM does.

Complexity Analysis. MM is equivalent to finding a *maximum bipartite matching* in a bipartite graph $G = (V = (X^{ask}, X^{bid}), E)$, where E only contains one edge for each pair of ask a and bid b if $p(a) \leq p(b)$. Let $n_a = |X^{ask}|$, $n_b = |X^{bid}|$, and n_{em} and n_{mm} are the numbers of matches got with EM and MM, respectively. MM runs in $O(n_a \log n_a) + O(n_b \log n_b) + O((n_{em})^2)$ time in the worst case, where $O(n_a \log n_a)$ and $O(n_b \log n_b)$ are the complexities of sorting asks and bids (e.g. merge sort), and $O((n_{em})^2)$ is that of the rest of MM. The worst case condition for MM is that $n_{em} = \min(n_a, n_b) - 1$ holds for all EMs of MM, unless $\min(n_a, n_b) \leq 1$. So we can rewrite the complexity of MM as $O(\max(n_a, n_b) \log \max(n_a, n_b) + \min(n_a, n_b)^2)$. As reference, the best known

worst-case performance bipartite matching algorithm is *the Hopcroft-Karp algorithm*, which runs in $O(|E|\sqrt{n_a + n_b})$, where $|E| \geq (n_{em})^2$ in our model, time in the worst case [8].

4 Properties of Maximal Matching

In this section, we analyse a number of key properties of MM, especially market liquidity. In the following, given a set of shouts X, we use $M_{EM}(X)$ and $M_{MM}(X)$ to indicate the matching got from EM and MM, respectively, and use $N_{EM}(X)$ and $N_{MM}(X)$ for the corresponding number of matches.

4.1 Maximizing the Number of Transactions

We prove that MM indeed maximizes the number of transactions. Before the proof, we first give two lemmas about the connection between MM and EM.

Lemma 1. *Given a set of shouts X, let X_{EM} and X_{MM} are all the shouts included in $M_{EM}(X)$ and $M_{MM}(X)$, respectively, then $X_{EM} \subseteq X_{MM}$.*

Proof. Since MM first finds all the matches that can be found with EM, then checks if unmatched shouts could be matched with the matched ones, and if so, adds extra matchable shouts in the matching by changing which ask and bid to match, but without removing any already matched shouts, so all shouts in $M_{EM}(X)$ are included in $M_{MM}(X)$. □

Lemma 2. *Given a set of shouts X, $N_{EM}(X) \leq N_{MM}(X) \leq 2 * N_{EM}(X)$.*

Proof. From Lemma 1, we can get $N_{MM}(X) \geq N_{EM}(X)$ because MM will return at least the number of matches returned from EM. Since in MM all the extra matchable shouts have to be matched with matched shouts in EM, so at most $N_{EM}(X)$ pairs of extra matchable shouts can be added. □

Theorem 1. *Given a set of shouts X, $N_{MM}(X)$ is maximal.*

Proof. Assume that $N_{MM}(X)$ is not maximal, i.e. there is at least one more pair of $a_{i>N_{MM}(X)}$ and $b_{j>N_{MM}(X)}$ (i and j can be different) from the sorted asks and bids that should be included in the result matching (assume that the index starts from 1). We also know $p(a_i) > p(b_j)$ because of the sort and the jump condition in Algorithm 3.1 (line 6). Thus a_i and b_j have to be matched with another bid and ask with index $\leq N_{EM}(X)$, respectively. Now we check two conditions of $N_{MM}(X)$: one is $N_{EM}(X) \leq N_{MM}(X) < 2 * N_{EM}(X)$ and the other is $N_{MM}(X) = 2 * N_{EM}(X)$.

1. If $N_{EM}(X) \leq N_{MM}(X) < 2 * N_{EM}(X)$, the recursive call of MM will end up with either $p(a_k) > p(b_{N_{MM}(X)-k+2})$ or $p(a_{N_{MM}(X)-k+2}) > p(b_k)$ for some integer $k \in [1, e+1]$, where $e = N_{MM}(X) - N_{EM}(X)$. If $p(a_k) > p(b_{N_{MM}(X)-k+2})$ for some k, based on the sort order, we also have $p(b_{N_{MM}(X)-k+2}) \geq p(b_j)$, so we get $p(a_k) > p(b_j)$, that is, b_j has to be

matched with some ask $a_{j'<k}$. However, each $a_{j'<k}$ has already been matched with another bid $b_{N_{MM}(X)-j'+1}$, and, based on the sort order, we know $p(b_{N_{MM}(X)-j'+1}) \leq p(b_{N_{MM}(X)-k+2})$, thus $p(a_k) > p(b_{N_{MM}(X)-j'+1})$, i.e. $b_{N_{MM}(X)-j'+1}$ cannot be matched with any ask ordered after a_{k-1}. If we want to match b_j, we have to remove some already matched bid. Similarly for $p(a_{N_{MM}(X)-k+2}) > p(b_k)$. Thus either a_i or b_j will not be matchable if $k = 1$, or some already matched shout(s) will be removed if $k > 1$, which contradicts the assumption.

2. If $N_{MM}(X) = 2 * N_{EM}(X)$, every shout with index $\leq N_{EM}(X)$ will be matched with another shout with index $> N_{EM}(X)$ in $M_{MM}(X)$. Thus we cannot match more shouts without losing matched ones in $M_{MM}(X)$, which again contradicts the assumption. □

4.2 Maximizing Sell/Buy-Volume

Apart from the number of transactions, sell-volume and buy-volume are also measures of the liquidity of a marketplace. We show that MM not only improves both of them compared with EM, but also maximizes buy-volume and minimizes sell-volume compared with any matching method that gives the same number of matches as MM.

Definition 2. *Given a matching M, the **sell-volume** is $\sum_{(a,b)\in M} p(a)$. The **buy-volume** is $\sum_{(a,b)\in M} p(b)$.*

Given a set of shouts X, we have $\sum_{(a,b)\in M_{MM}(X)} p(a) \geq \sum_{(a,b)\in M_{EM}(X)} p(a)$ and $\sum_{(a,b)\in M_{MM}(X)} p(b) \geq \sum_{(a,b)\in M_{EM}(X)} p(b)$ because of Lemma 1. So MM will improve, if possible, sell-volume and buy-volume compared with EM.

Theorem 2. *Given a set of shouts X, $\forall M \in \{M : |M(X)| = N_{MM}(X)\}$ $\sum_{(a,b)\in M_{MM}(X)} p(a) \leq \sum_{(a,b)\in M(X)} p(a) \wedge \sum_{(a,b)\in M_{MM}(X)} p(b) \geq \sum_{(a,b)\in M(X)} p(b)$.*

Proof. Given a set of shouts X, from the MM algorithm, we know that the first $N_{MM}(X)$ asks in ascending order and the first $N_{MM}(X)$ bids in descending order are matched. Thus the corresponding sell-volume and buy-volume will be minimal and maximal, respectively, given that the number of matches is $N_{MM}(X)$. □

4.3 Maximizing Profit

In this section, we give the definition of the auctioneer's profit and analyse the impact of MM on that profit. Especially, we prove that MM maximizes the auctioneer's profit compared with any other matching method that matches the same number of pairs as MM.

Definition 3. *Given matching M, the auctioneer's profit is the difference between the sum of the price of all bids and that of all asks in M:*

$$U(M) = \sum_{(a,b)\in M} p(b) - \sum_{(a,b)\in M} p(a)$$

Theorem 3. *Given a set of shouts X, $U(M_{MM}(X)) \leq U(M_{EM}(X))$. Furthermore, $\forall M \in \{M : |M(X)| = N_{MM}(X)\} U(M_{MM}(X)) \geq U(M(X))$.*

Proof. Given a set of shouts X, let $m = N_{EM}(X)$ and $n = N_{MM}(X)$. From Lemma 2 we know $m \leq n$. From Lemma 1, if $m = n$, we have $U(M_{MM}(X)) = U(M_{EM}(X))$. If $m < n$, then

$$U(M_{EM}(X)) = \sum_{i=1}^{m} (p(b_i) - p(a_i))$$
$$U(M_{MM}(X)) = \sum_{i=1}^{m} (p(b_i) - p(a_i)) + \sum_{i=m+1}^{n} (p(b_i) - p(a_i))$$

where b_i means the ith bid from X^{bid} in descending order and a_i is the ith ask from X^{ask} in ascending order w.r.t. their price. Since $p(b_i) \geq p(a_i)$ when $i \leq m$, and $p(b_i) < p(a_i)$ when $i > m$, we get $\sum_{i=1}^{m} (p(b_i) - p(a_i)) \geq 0$ and $\sum_{i=m+1}^{n} (p(b_i) - p(a_i)) < 0$. Thus $U(M_{MM}(X)) \leq U(M_{EM}(X))$.

Given the number of matches $N_{MM}(X)$, we can easily get $U(M_{MM}(X))$ is maximal from Theorem 2. □

Although the auctioneer's profit with MM might be less than that with EM in the short term, the lost profit is actually used in liquidity maximization. High liquidity attracts traders, which will recursively increase liquidity and also lead to increased profit in the long term. We will show this from the experiments in Sect. 5.

4.4 Maximizing Social Welfare

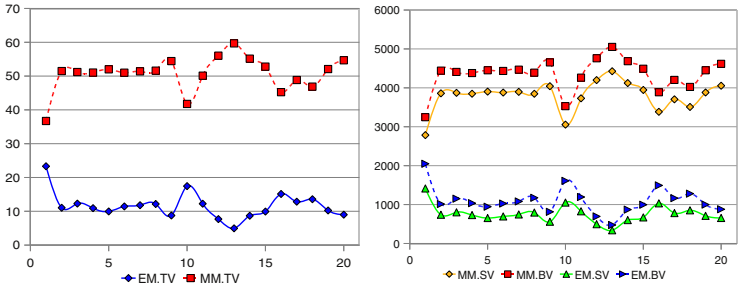
Maximizing social welfare means maximizing the sum of traders' valuations of the item they are holding, i.e. matched buyers and unmatched sellers. Given bids B' that are matched and asks A' that are unmatched, then the term that we wish to maximize is: $\sum_{b \in B'} p(b) + \sum_{a \in A'} \delta p(a)$, where $\delta \in [0, 1]$ is a discount factor of the valuation of the goods that are not matched/traded. This is reasonable because, for instance, some commodities have an expiry date, e.g. ice cream and flowers, and there might be an inventory cost for un-traded goods.

Let B_{MM} and B_{EM} are the matched bids in MM and EM, respectively, we know from Lemma 1 that $B_{MM} \supseteq B_{EM}$, so we have $\sum_{b \in B_{MM}} p(b) \geq \sum_{b \in B_{EM}} p(b)$. If $\delta = 0$, MM maximizes social welfare. The proof is similar to the proof of Theorem 2.

5 Experimental Results

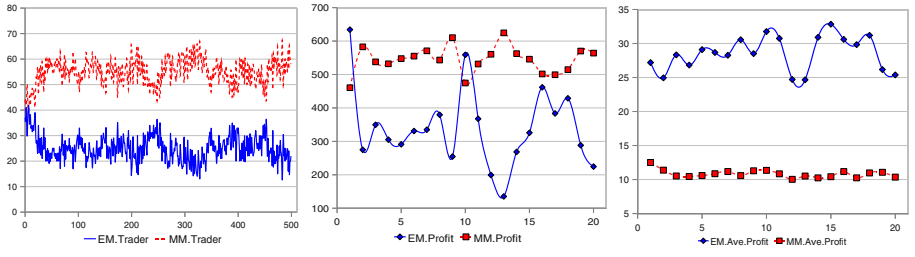
We have integrated the EM and MM specified in Sect. 3 in *jackaroo*[4] which is a successful market under the CAT Tournament (Trading Agent Competition Market Design) platform. A CAT game is a simulation of exchange markets, and consists of buyers, sellers, and specialists. Each specialist operates and sets the rules for a single exchange market, and traders buy and sell goods in one of the available markets. In the CAT tournaments, the buyers and sellers are provided by the organizers, whereas specialists are designed by the entrants [9].

[4] Achieved 3rd, 1st, and 2nd in CAT Tournament 2008, 2009, and 2010, respectively.



(a) Transaction Volume (TV) (b) Buy-/Sell-volume (BV/SV)

Fig. 2. Market Liquidity



(a) Trader Distribution (b) Profit (c) Average Profit/Match

Fig. 3. Trader Distribution and Auctioneer's Profit

In our experiments, we use two markets (two variations of *jackaroo*) that use the same rules except that one uses EM and the other uses MM, and 80 profit-seeking traders with different intelligences to simulate a real market situation. We compare these two markets in 500 virtual days. In each day, each trader chooses only one market to exchange goods, and depending on the profit a trader got in that market, the trader might move to the other market on the next day.

Figures 2 and 3 show the main results related to our analysed properties in Sect. 4. Please note that Fig. 3 (a) shows information for each day, the rest use the average value of every 25 days. From Fig. 2 we can see that the transaction volume with MM is about 5 times that with EM, and both the buy-volume and sell-volume of the market with MM are about 4 times those of the EM market. As we mentioned in Sect. 4, good market liquidity will attract traders to the marketplace. This can be observed from the experiments. Figure 3 (a) shows the trader distribution in each day: in the very beginning, traders are equally distributed, but after 50 days, the MM market attracted most traders and it could maintain this until day 500. Because of trader attraction, the auctioneer's profit in the MM market is actually higher than that gained from the EM market (see Fig. 3 (b)), though the average profit for each transaction in the MM market is much smaller than that in the EM market (see Fig. 3 (c)). Note that, because the CAT Tournament platform was designed specifically for Trading

Agent Competition, it may not be the perfect testing environment for matching policies. Nonetheless, our experiments have provided a positive support to our theoretical results.

6 Conclusion

We have developed a new matching approach, called maximal matching, for a double auction market. We compared maximal matching with the most studied matching algorithm, equilibrium matching, and analysed its properties. The most distinctive property of maximal matching is liquidity maximization, which is a key factor in measuring the success of a marketplace. We also showed the advantages of maximal matching from experiments by using the CAT Tournament platform.

It is easy to see that a double auction mechanism with maximal matching is not necessarily incentive compatible. The essential feature of incentive compatibility is that the clearing price of a match does not depend on the bidding prices of the match. Maximal matching cannot guarantee this because the intersection of the clearing price ranges of all matches might be empty, while equilibrium matching can have incentive compatibility depending on how the uniform clearing price is chosen [4,5]. However, incentive compatibility is not compatible with most other desirable properties and is also very hard to achieve, especially in *dynamic/online double auction* (e.g. stock exchanges), where shouts are coming and leaving over time and there is more than one matching to search sequentially [3,10].

References

1. Friedman, D., Rust, J.: The Double Auction Market: Institutions, Theories, And Evidence. Westview Press, Boulder (March 1993)
2. Shubik, M.: A double auction market: Teaching, experiment, and theory. Simul. Gaming 36(2), 166–182 (2005)
3. Blum, A., Sandholm, T., Zinkevich, M.: Online algorithms for market clearing. J. ACM 53(5), 845–879 (2006)
4. McAfee, P.R.: A dominant strategy double auction. Journal of Economic Theory 56(2), 434–450 (1992)
5. Wurman, P.R., Walsh, W.E., Wellman, M.P.: Flexible double auctions for electionic commerce: theory and implementation. Decis. Support Syst. 24(1), 17–27 (1998)
6. Dash, R.K., Jennings, N.R., Parkes, D.C.: Computational-Mechanism design: A call to arms. IEEE Intelligent Systems 18(6), 40–47 (2003)
7. Nagle, T., Holden, R.: The Strategy and Tactics of Pricing: a guide to profitable decision making. Prentice Hall, Englewood Cliffs (November 2001)
8. Hopcroft, J.E., Karp, R.M.: A n5/2 algorithm for maximum matchings in bipartite. In: SWAT 1971: Proceedings of the 12th Annual Symposium on Switching and Automata Theory, Washington, pp. 122–125. IEEE Computer Society, Los Alamitos (1971)
9. Cai, K., Gerding, E., Mcburney, P., Niu, J., Parsons, S., Phelps, S.: Overview of cat: A market design competition version 2.0 (2009)
10. Bredin, J., Parkes, D.C., Duong, Q.: Chain: a dynamic double auction framework for matching patient agents. J. Artif. Int. Res. 30(1), 133–179 (2007)

Author Index