Byeong-Ho Kang
Debbie Richards (Eds.)

Knowledge Management and Acquisition for Smart Systems and Services

LNAI 6232

11th International Workshop, PKAW 2010
Daegu, Korea, August/September 2010
Proceedings

Lecture Notes in Artificial Intelligence 6232

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

Subseries of Lecture Notes in Computer Science

Byeong-Ho Kang Debbie Richards (Eds.)

Knowledge Management and Acquisition for Smart Systems and Services

11th International Workshop, PKAW 2010
Daegu, Korea, August 20 - September 3, 2010
Proceedings

Springer

Volume Editors

Byeong-Ho Kang
University of Tasmania
School of Computing and Information Systems
Launceston, TAS 7250 Tasmania, Australia
E-mail: bhkang@utas.edu.au

Debbie Richards
Macquarie University
Department of Computing, Faculty of Science
Sydney, NSW, 2109, Australia
E-mail: richards@ics.mq.edu.au

Library of Congress Control Number: 2010931852

CR Subject Classification (1998): I.2, H.3, H.4, H.5, C.2, J.1
LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-642-15036-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-15036-4 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

The 11th International Workshop on Knowledge Management and Acquisition for Smart Systems and Services (PKAW 2010) has provided a forum for the past two decades for researchers and practitioners working in the area of machine intelligence. PKAW covers a spectrum of techniques and approaches to implement smartness in IT applications. As evidenced in the papers in this volume, machine intelligence solutions incorporate many areas of AI such as ontological engineering, agent-based technology, robotics, image recognition and the Semantic Web as well as many other fields of computing such as software engineering, security, databases, the Internet, information retrieval, language technology and game technology.

PKAW has evolved to embrace and foster advances in theory, practice and technology not only in knowledge acquisition and capture but all aspects of knowledge management including reuse, sharing, maintenance, transfer, merging, reconciliation, creation and dissemination. As many nations strive to be knowledge economies and organizations seek to maximize their knowledge assets and usage, solutions to handle the complex task of knowledge management are more important than ever. This volume contributes towards this goal.

This volume seeks to disseminate the latest solutions from the International Workshop on Knowledge Management and Acquisition for Smart Systems and Services (PKAW 2010) held in Daegu, Korea during August 30–31, 2010 in conjunction with the Pacific Rim International Conference on Artificial Intelligence (PRICAI 2010). The workshop received 94 submissions. From these, we accepted 26 papers (28%) for full presentations. All papers were blind reviewed by at least two members of the PKAW/PRICAI Program Committee. The papers demonstrate a balance of theoretical, technical and application–driven research, many papers incorporating all three foci. Approximately half the papers reflect the increasing use of KA methods for application areas such as mobile computing, Internet/WWW and game/multimedia areas.

The Workshop Co-chairs would like to thank all those who were involved with PKAW 2010 including the PRICAI 2010 Organizing Committee, PKAW Program Committee members, those who submitted papers and reviewed them and of course the authors, presenters and attendees. We warmly invite you to participate in PKAW 2012, anticipated to be held in conjunction with PRICAI 2012.

July 2010

Byeong Ho Kang
Debbie Richards

Organization

General Co-chair

Paul Compton	University of New South Wales, Australia
Hiroshi Motoda	Osaka University, Japan

Program Co-chair

Byeong Ho Kang	University of Tasmania, Australia
Debbie Richard	Macquarie University, Australia

Local Chair

Tai- Hoon Kim	Hannam University, South Korea

Publicity Chair

Yangsok Kim	University of New South Wales, Australia

Program Committee

Quan Bai	CSIRO, Australia
Rodrigo Martinez-Bejar	Universidad de Murcia, Spain
Ivan Bindoff	University of Tasmania, Australia
Nguyen Dang Binh	Graz University of Technology, Austria
Joachim Baumeister	University of Würzburg, Germany
Paul Compton	University of New South Wales, Australia
Richard Dazeley	University of Ballarat, Australia
Peter Eklund	University of Wollongong, Australia
Jesualdo Tomas Fernadez-Breis	Universidad de Murcia, Spain
Windy Gambeta	Institut Teknologi Bandung, Indonesia
Francisco Garcia-Sanchez	Universidad de Murcia, Spain
Aditya K. Ghose	University of Wollongong, Australia
Fabrice Guillet	L'Universite de Nantes, France
Udo Hahn	Jena University, Germany
Ray Hashemi	Armstrong Atlantic State University, USA
Achim Hoffmann	University of New South Wales, Australia
Noriaki Izumi	Cyber Assist Research Center, AIST, Japan
Byeong Ho Kang	University of Tasmania, Australia

Mihye Kim	Catholic University of Daegu, South Korea
Seok Soo Kim	Hannam University, South Korea
Tae Hoon Kim	Hannam University, South Korea
Yang Sok Kim	University of New South Wales, Australia
Maria R. Lee	Shih Chien University, Taiwan
Huan Liu	Arizona State University, USA
Tim Menzies	NASA, USA
Kyong Ho Min	University of New South Wales, Australia
Toshiro Minami	Kyushu Institute of Information Sciences & Kyushu University, Japan
Hiroshi Motota	Osaka University, Japan
Masayuki Numao	Osaka University, Japan
Kouzou Ohara	Aoyama Gakuin University, Japan
Ulrich Reimer	University of Applied Science St. Gallen, Switzerland
Debbie Richards	Macquarie University, Australia
Young Ju Rho	Korea Polytechnic University, South Korea
Takao Terano	University of Tsukuba, Japan
Shusaku Tsumoto	Shimane University, Japan
Abdul Satar	Griffith University, Australia
Hendra Suryanto	Institute of Analytics Professionals of Australia (IAPA), Australia
Rafael Valencia-Garcia	Universidad de Murcia, Spain
Bay Vo	Ho Chi Minh City University of Technology, Vietnam
Takashi Washio	Osaka University, Japan
Shuxiang Xu	University of Tasmania, Australia
Jung Jin Yang	The Catholic University of Korea, South Korea
Tatjana Zrimec	University of New South Wales, Australia

Table of Contents

Machine Learning

A Graph-Based Projection Approach for Semi-supervised Clustering 1
Tetsuya Yoshida and Kazuhiro Okatani

Self-organisation in an Agent Network via Multiagent Q-Learning 14
Dayong Ye, Minjie Zhang, Quan Bai, and Takayuki Ito

Improving Trading Systems Using the RSI Financial Indicator and Neural Networks 27
Alejandro Rodríguez-González, Fernando Guldrís-Iglesias, Ricardo Colomo-Palacios, Juan Miguel Gomez-Berbis, Enrique Jimenez-Domingo, Giner Alor-Hernandez, Rubén Posada-Gomez, and Guillermo Cortes-Robles

Balanced Student Partitioning to Promote Effective Learning: Applications in an International School 38
Wenbin Zhu, Hu Qin, Andrew Lim, and Zhou Xu

Data Mining

Laban-Based Motion Rendering for Emotional Expression of Human Form Robots 49
Megumi Masuda, Shohei Kato, and Hidenori Itoh

Self-supervised Mining of Human Activity from CGM 61
Nguyen Minh The, Takahiro Kawamura, Hiroyuki Nakagawa, Yasuyuki Tahara, and Akihiko Ohsuga

Data Mining Using an Adaptive HONN Model With Hyperbolic Tangent Neurons 73
Shuxiang Xu

Business Intelligence for Delinquency Risk Management via Cox Regression 82
Sung Ho Ha and Eun Kyoung Kwon

Knowledge Engineering & Ontology

An Ontology-Based Adaptive Learning System to Enhance Self-directed Learning 91
Mihye Kim and Sook-Young Choi

Context-Aware Service Framework for Decision-Support Applications Using Ontology-Based Modeling 103
Giovanni Cagalaban and Seoksoo Kim

A Disaster Management Metamodel (DMM) Validated 111
Siti Hajar Othman and Ghassan Beydoun

Another Investigation of an Interontologia between Chinese Lexical Systems and Roget's Thesaurus 126
Sang-Rak Kim, Jae-Gun Yang, and Jae-Hak J. Bae

Incremental Knowledge Acquisition

Incremental Knowledge Acquisition Using Generalised RDR for Soccer Simulation 135
Angela Finlayson and Paul Compton

Incremental System Engineering Using Process Networks 150
Avishkar Misra, Arcot Sowmya, and Paul Compton

RDRCE: Combining Machine Learning and Knowledge Acquisition 165
Han Xu and Achim Hoffmann

Simulated Assessment of Ripple Round Rules 180
Ivan Bindoff and Byeong Ho Kang

The Ballarat Incremental Knowledge Engine 195
Richard Dazeley, Philip Warner, Scott Johnson, and Peter Vamplew

KA Applications in Internet and Mobile Computing

Finding Relation between PageRank and Voter Model 208
Takayasu Fushimi, Kazumi Saito, Masahiro Kimura, Hiroshi Motoda, and Kouzou Ohara

Mobile Sync-application for Life Logging and High-Level Context Using Bayesian Network 223
Tae-min Jung, Young-Seol Lee, and Sung-Bae Cho

Consensus Clustering and Supervised Classification for Profiling Phishing Emails in Internet Commerce Security 235
Richard Dazeley, John L. Yearwood, Byeong H. Kang, and Andrei V. Kelarev

People Recommendation Based on Aggregated Bidirectional Intentions in Social Network Site 247
Yang Sok Kim, Ashesh Mahidadia, Paul Compton, Xiongcai Cai, Mike Bain, Alfred Krzywicki, and Wayne Wobcke

Visualising Intellectual Structure of Ubiquitous Computing 261
Maria R. Lee and Tsung Teng Chen

Acquiring Expected Influence Curve from Single Diffusion Sequence 273
Yuya Yoshikawa, Kazumi Saito, Hiroshi Motoda, Kouzou Ohara, and Masahiro Kimura

KA Applications in Multimedia and Games

Automatic Speech-Based Classification of Gender, Age and Accent 288
Phuoc Nguyen, Dat Tran, Xu Huang, and Dharmendra Sharma

MMG: A Learning Game Platform for Understanding and Predicting Human Recall Memory 300
Umer Fareed and Byoung-Tak Zhang

Efficient Bulk-Insertion for Content-Based Video Indexing 310
Narissa Onkhum and Juggapong Natwichai

Author Index 323

A Graph-Based Projection Approach for Semi-supervised Clustering

Tetsuya Yoshida and Kazuhiro Okatani

Grad. School of Information Science and Technology,
Hokkaido University
N-14 W-9, Sapporo 060-0814, Japan
{yoshida,okatani}@meme.hokudai.ac.jp

Abstract. This paper proposes a graph-based projection approach for semi-supervised clustering based on pairwise relations among instances. In our approach, the entire data is represented as an edge-weighted graph with the pairwise similarities among instances. Graph representation enables to deal with two kinds of pairwise constraints as well as pairwise similarities over the same unified representation. Then, in order to reflect the pairwise constraints on the clustering process, the graph is modified by contraction in graph theory and graph Laplacian in spectral graph theory. By exploiting the constraints as well as similarities among instances, the entire data are projected onto a subspace via the modified graph, and data clustering is conducted over the projected representation. The proposed approach is evaluated over several real world datasets. The results are encouraging and indicate the effectiveness of the proposed approach.

1 Introduction

Data clustering, also called unsupervised classification, is a method of creating groups of objects, or clusters, in such a way that objects in one cluster are very similar and objects in different clusters are quite distinct. Clustering is one of the most frequently performed analysis [12]. For example, in web activity logs, clusters can indicate navigation patterns of different user groups. Another direct application could be clustering of gene expression data so that genes within a same group evinces similar behavior.

Recently, semi-supervised clustering, learning from a combination of labeled and unlabeled data, has been intensively studied in data mining and machine learning communities [3]. One of the reasons is that, small amount of additional information such as labeled instances or pairwise instance constraints are relatively easy to collect and still can improve the performance significantly. Various research efforts have been conducted on semi-supervised clustering. Among them, [16] proposed a feature projection approach for handling high-dimensional data, and reported that it outperformed other existing methods. However, although pairwise constraints among instances are dealt with, pairwise relations among instances are not explicitly utilized.

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 1–13, 2010.

This paper proposes a graph-based projection approach for semi-supervised clustering. When the similarities among instances are specified, by connecting each pair of instances with an edge, the entire data can be represented as an edge-weighted graph. Graph representation enables to deal with two kinds of pairwise constraints as well as pairwise similarities over the same unified representation. Then, the graph is modified by contraction in graph theory [8] and graph Laplacian in spectral graph theory [4,17] to reflect the pairwise constraints.

Representing the relations (constraints and similarities) among instances as an edge-weighted graph and modifying the graph structure based on the specified constraints enable to enhancing semi-supervised clustering based on the pairwise relations among instances. The entire data are projected onto a subspace which is constructed via the modified graph, and clustering is conducted over the projected representation. Although the proposed approach utilizes graph Laplacian as in [2], it differs since pairwise constraints for semi-supervised clustering are also utilized for constructing the projected representation. The proposed approach is evaluated over several real world datasets. The results are encouraging and indicate the effectiveness of the proposed approach in terms of accuracy and running time.

1.1 Related Work

In general, clustering methods are divided into two approaches: hierarchical methods and partitioning methods [12]. Hierarchical methods construct a cluster hierarchy, or a tree of clusters (called a dendrogram), whose leaves are the data points and whose internal nodes represent nested clusters of various sizes. On the other hand, partitioning methods return a single partition of the entire data under a fixed parameters (number of clusters, thresholds, etc.). Each cluster can be represented by its centroid [11] or by one of its objects located near its center [14].

When similarities among the pairs of instances can be estimated, the entire data can be represented as an edge-weighted graph. Several graph-theoretic clustering methods have been proposed. Various graph-theoretic clustering approaches tries to find subsets of vertices in a graph based on the edges among the vertices. Several methods utilizes graph coloring techniques [10,9]. Other methods are based on the flow or cut in graph, such as spectral clustering [17].

Semi-supervised clustering methods can be categorized into three approaches: constraint-based, distance-based, and hybrid approaches [16]. The constraint-based approach tries to guide the clustering process with the specified pairwise instance constraints [18]. The distance-based approach utilizes metric learning techniques to acquire the distance measure during the clustering process based on the specified pairwise instance constraints [19,13]. The hybrid approach combines these two approaches under a probabilistic framework [1].

Organization. Section 2 explains the details of the proposed approach for clustering under pairwise constraints. Section 3 reports the evaluation of the proposed approach over several benchmark datasets. Section 4 summarizes our contributions and suggests future directions.

2 Graph-Based Semi-supervised Clustering

2.1 Preliminaries

Let $\boldsymbol{X}$ be a set of instances. For a set $\boldsymbol{X}$, $|\boldsymbol{X}|$ represents its cardinality.

A graph $G(\boldsymbol{V}, \boldsymbol{E})$ consists of a finite set of vertices $\boldsymbol{V}$, a set of edges $\boldsymbol{E}$ over $\boldsymbol{V} \times \boldsymbol{V}$. The set $\boldsymbol{E}$ can be interpreted as representing a binary relation on $\boldsymbol{V}$. A pair of vertices (v_i, v_j) is in the binary relation defined by a graph $G(\boldsymbol{V}, \boldsymbol{E})$ if and only if the pair $(v_i, v_j) \in \boldsymbol{E}$.

An edge-weighted graph $G(\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W})$ is defined as a graph $G(\boldsymbol{V}, \boldsymbol{E})$ with the weight on each edge in $\boldsymbol{E}$. When $|\boldsymbol{V}| = n$, the weights in $\boldsymbol{W}$ can be represented as an n by n matrix $\mathbf{W}$ [1], where w_{ij} in $\mathbf{W}$ stands for the weight on the edge for the pair $(v_i, v_j) \in \boldsymbol{E}$. We set $w_{ij} = 0$ for pairs $(v_i, v_j) \notin \boldsymbol{E}$. In addition, we assume that $G(\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W})$ is an undirected, simple graph without self-loop. Thus, the weight matrix $\mathbf{W}$ is symmetric and its diagonal elements are zeros.

2.2 Problem Setting

When the semi-supervised information about clustering is represented as a set of constraints, the *semi-supervised clustering* problem is described as follows.

Problem 1 (Semi-Supervised Clustering). For a given set of data $\boldsymbol{X}$ and specified constraints, find a partition (a set of clusters) $\boldsymbol{T} = \{t_1, \ldots, t_k\}$ which satisfies the specified constraints.

There can be various forms of constraints. Based on the previous work [18,19,16,13], we consider the following two kinds of constraints in this paper: must-link constraints and cannot-link constraints.

Definition 1 (Pairwise Constraints). *For a given set of data $\boldsymbol{X}$ and a partition (a set of clusters) $\boldsymbol{T} = \{t_1, \ldots, t_k\}$, must-link constraints $\boldsymbol{C}_{ML}$ and cannot-link constraints $\boldsymbol{C}_{CL}$ are sets of pairs such that:*

$$\exists (x_i, x_j) \in \boldsymbol{C}_{ML} \Rightarrow \exists t \in \boldsymbol{T}, (x_i \in t \wedge x_j \in t) \tag{1}$$

$$\exists (x_i, x_j) \in \boldsymbol{C}_{ML} \Rightarrow \exists t_a, t_b \in \boldsymbol{T}, t_a \neq t_b, (x_i \in t_a \wedge x_j \in t_b) \tag{2}$$

$\boldsymbol{C}_{ML}$ specifies the pairs of instances in the same cluster, and $\boldsymbol{C}_{CL}$ specifies the pairs of instances in different clusters.

2.3 A Graph-Based Approach

By assuming that some similarity measure for the pairs of instances $\boldsymbol{X}$ is specified, we propose a graph-based approach for constrained clustering problem. Based on the similarities, the entire data $\boldsymbol{X}$ can be represented as an edge-weighted graph $G(\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W})$ where w_{ij} represents the similarity between a pair (x_i, x_j). Since each data object $x \in \boldsymbol{X}$ corresponds to a vertex $v \in \boldsymbol{V}$ in G, we

[1] A bold italic symbol $\boldsymbol{W}$ denotes a set, while a bold symbol $\mathbf{W}$ denotes a matrix.

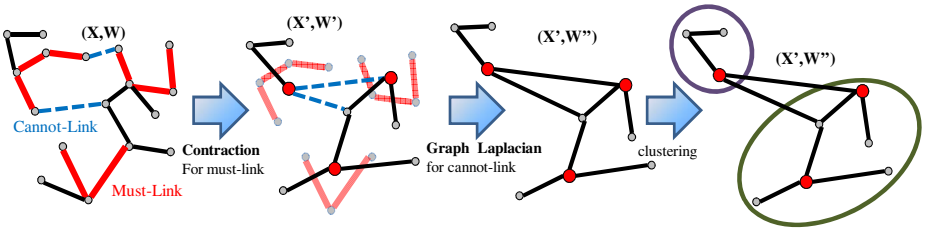


Fig. 1. Overview of graph-based projection approach

abuse the symbol $\boldsymbol{X}$ to denote the set of vertices in G in the rest of the paper. We assume that all w_{ij} is non-negative.

Definition 1 specifies two kinds of constraints. For $\boldsymbol{C}_{ML}$, we propose a method based on graph contraction in graph theory [8] and treat it as hard constraints (Sections 2.4); for $\boldsymbol{C}_{CL}$, we propose a method based on graph Laplacian in spectral graph theory [4,17] and treat it as soft constraints under the optimization framework (Section 2.5). The overview of our approach is illustrated in Fig. 1.

2.4 Graph Contraction for Must-Link Constraints

For must-link constraints $\boldsymbol{C}_{ML}$ in eq.(1), the transitive law holds; *i.e.*, for any two pairs (x_i, x_j) and $(x_j, x_l) \in \boldsymbol{C}_{ML}$, x_i and x_l should also be in the same cluster. In order to enforce the transitive law in $\boldsymbol{C}_{ML}$, we propose to utilize graph contraction [8] to the graph G for a data set $\boldsymbol{X}$.

Definition 2 (Contraction). *Let e=(x_i, x_j) be an edge of a graph G = ($\boldsymbol{X}$, $\boldsymbol{E}$). By G/e, we denote the graph ($\boldsymbol{X}$',$\boldsymbol{E}$') obtained from G by contracting the edge e into a new vertex x_e, where:*

$$\boldsymbol{X}' = (\boldsymbol{X} \backslash \{x_i, x_j\}) \cup \{x_e\} \tag{3}$$

$$\boldsymbol{E}' = \{(u, v) \in \boldsymbol{E} | \{u, v\} \cap \{x_i, x_j\} = \phi\} \cup \{(x_e, u) | (x_i, u) \in \boldsymbol{E} \backslash \{e\} \ or \ (x_j, u) \in \boldsymbol{E} \backslash \{e\}\} \tag{4}$$

By contracting the edge e into a new vertex x_e, it becomes adjacent to all the former neighbors of x_i and x_j. Recursive application of contraction guarantees that the transitive law in $\boldsymbol{C}_{ML}$ is sustained in the cluster assignment.

In our approach, the original data $\boldsymbol{X}$ is represented as an edge-weighted graph $G(\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{W})$. Thus, after contracting an edge e=$(x_i, x_j) \in \boldsymbol{C}_{ML}$ into the vertex x_e, it is necessary to define the weights in the contracted graph G/e. The weights in G represent the similarities among vertices. The original similarities should at least be sustained after contracting an edge in $\boldsymbol{C}_{ML}$, since must-link constraints are for enforcing the similarities, not for reducing.

Based on the above observation, we propose to define the weights in G/e as:

$$w(x_e, u) = \max(w(x_i, u), w(x_j, u)) \quad (x_i, u) \in \boldsymbol{E} \ or (x_j, u) \in \boldsymbol{E} \tag{5}$$

$$w(u, v) = w(u, v) \quad otherwise \tag{6}$$

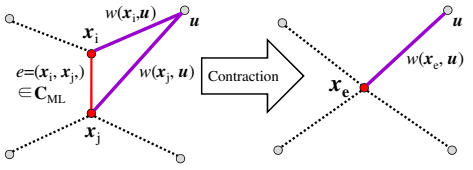


Fig. 2. Contraction for must-link constraints

In eq.(5), the function max realizes the above requirement, and guarantees the *monotonic* (non-decreasing) properties of similarities (weights) after contraction.

As illustrated in Fig. 2, for each pair of edges in $\boldsymbol{C}_{ML}$, we apply graph contraction and define weights in the contracted graph based on eqs.(5) and (6). This results in modifying the original graph G into $G^{'}(\boldsymbol{X}^{'}, \boldsymbol{E}^{'}, \boldsymbol{W}^{'})$, where $n^{'} = |\boldsymbol{X}^{'}|$.

2.5 Graph Laplacian for Cannot-Link Constraints

To reflect cannot-link constraints in the clustering process, we formalize the clustering under the constraints as an optimization problem, and consider the minimization of the following objective function:

$$J = \frac{1}{2}\{\sum_{i,j} w_{ij}^{'}||f_i - f_j||^2 - \lambda \sum_{u,v \in C_{CL}^{'}} w_{uv}^{'}||f_u - f_v||^2\} \tag{7}$$

where i and j sum over the vertices in the contracted graph $G^{'}$, and $\boldsymbol{C}_{CL}^{'}$ stands for the cannot-link constraints in $G^{'}$. f_i stands for the value assigned for data x_i, and $\lambda \in [0, 1]$ is a hyper-parameter. The first term corresponds to the smoothness of the assigned values in spectral graph theory, and the second term represents the influence of $\boldsymbol{C}_{CL}^{'}$ in optimization. Note that by setting $\lambda \in [0, 1]$, the objective function in (7) is guaranteed to be a convex function. Also, $\boldsymbol{C}_{CL}^{'}$ is treated as soft constraints under the optimization framework in our current approach.

From the above objective function in eq.(7), we can derive the following unnormalized graph Laplacian $\mathbf{L}^{''}$ which incorporates $\boldsymbol{C}_{CL}$:

$$\begin{aligned} J &= \frac{1}{2}\{\sum_{i,j} w_{ij}^{'}||f_i - f_j||^2 - \lambda \sum_{u,v \in C_{CL}^{'}} w_{uv}^{'}||f_u - f_v||^2\} \\ &= \boldsymbol{f}^t \mathbf{D}^{''} \boldsymbol{f} - \boldsymbol{f}^t \mathbf{W}^{''} \boldsymbol{f} \qquad (8) \\ &= \boldsymbol{f}^t \mathbf{L}^{''} \boldsymbol{f} \qquad (9) \end{aligned}$$

where $\boldsymbol{f}^t$ stands for the transposition of vector $\boldsymbol{f}$, $\odot$ stands for the Hadamard product (element-wise multiplication) of two matrices, and:

$$(\mathbf{C}^{'})_{uv} = \begin{cases} 1 & (x_u, x_v) \in C_{CL}^{'} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$\mathbf{W}^c = \mathbf{C}^{'} \odot \mathbf{W}^{'}, \quad \mathbf{W}^{''} = \mathbf{W}^{'} - \lambda \mathbf{W}^c \tag{11}$$

$$d_i = \sum_{j=1}^{n^{'}} w_{ij}^{'}, \quad d_i^c = \sum_{j=1}^{n^{'}} w_{ij}^c \tag{12}$$

$$\mathbf{D}^{''} = diag(d_1^{''}, \ldots, d_{n^{'}}^{''}), \quad d_i^{''} = d_i - \lambda d_i^c \tag{13}$$

$$\mathbf{L}^{''} = \mathbf{D}^{''} - \mathbf{W}^{''} \tag{14}$$

The above process amounts to modifying the representation of the graph $G^{'}$ into $G^{''}$, where the modified weights $\mathbf{W}^{''}$ are defined in eq.(11). Thus, as illustrated in Fig. 1, the proposed approach modifies the original graph G into $G^{'}$ with must-link constraints and then into $G^{''}$ with cannot-link constraints and similarities.

It is known that some "balancing" of clusters is required for obtaining meaningful results [17]. Based on eqs.(12) and (14), we propose the following normalized objective function:

$$J_{sym} = \sum_{i,j} w_{ij}^{''} || \frac{f_i}{\sqrt{d_i^{''}}} - \frac{f_j}{\sqrt{d_j^{''}}} ||^2 \tag{15}$$

over the graph $G^{''}$. Minimizing J_{sym} in eq.(15) amounts to solving the generalized eigen-problem $\mathbf{L}^{''}\boldsymbol{h} = \alpha \mathbf{D}^{''}\boldsymbol{h}$, where $\boldsymbol{h}$ corresponds to the generalized eigenvector and α corresponds to the eigenvalue.

2.6 The Algorithm

The proposed graph-based semi-supervised clustering method GBSSC is summarized in Algorithm 1. Lines 1 to 3 create the contracted graph $G^{'}$. Lines 4 to 6 conduct the minimization of J_{sym} in eq.(15), which is represented as the normalized graph Laplacian $\mathbf{L}_{sym}^{''}$ at line 5. These correspond to the spectral embedding

Algorithm 1. graph-based semi-supervised clustering GBSSC

Require: $G(\boldsymbol{X}, \boldsymbol{E}, \boldsymbol{W})$; *//an edge-weighted graph*
Require: $\boldsymbol{C}_{ML}$; *//must-link constraints*
Require: $\boldsymbol{C}_{CL}$; *//cannot-link constraints*
Require: l; *//the number of dimensions of the subspace*
Require: k; *//the number of clusters*

1: **for** each $e \in C_{ML}$ **do**
2: contract e and create the contracted graph G/e;
3: **end for**// *Let* $G^{'}(\boldsymbol{X}^{'}, \boldsymbol{E}^{'}, \boldsymbol{W}^{'})$ *be the contracted graph.*
4: create $\mathbf{C}_{uv}^{'}, \mathbf{W}^c, \mathbf{W}^{''}, \mathbf{D}^{''}$ as eqs.(10) $\sim$ (13).
5: $\mathbf{L}_{sym}^{''} = \mathbf{I} - \mathbf{D}^{''-\frac{1}{2}}\mathbf{W}^{''}\mathbf{D}^{''-\frac{1}{2}}$
6: Find l eigenvectors $\mathbf{H} = \{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_l\}$ for $\mathbf{L}_{sym}^{''}$, with the smallest non-zero eigenvalues.
7: Conduct clustering of data which are represented as $\mathbf{H}$ and construct clusters.
8: **return** clusters

of $\boldsymbol{X}$ onto the subspace spanned by $\mathbf{H} = \{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_l\}$ [2]. Note that pairwise constraints for semi-supervised clustering are also utilized on the construction of the embedded representation in our approach and thus differs from [2]. Some clustering method is applied to the data at line 7 and the constructed clusters are returned. Currently spherical kmeans (skmeans)[2] [7] is utilized at line 7.

3 Evaluations

3.1 Experimental Settings

Datasets. Based on the previous work [6,16], we evaluated it on 20 Newsgroup data (20NG)[3] and TREC datasets[4]. Clustering of these datasets corresponds to document clustering, and each document is represented as the standard vector space model based on the occurrences of terms. Note that the proposed method is generic and not specific to document clustering. Since the number of terms are huge in general, these are high-dimensional sparse datasets.

Table 1. TREC datasets

dataset	# attr.	#classes	#data
tr11	6429	9	414
tr12	5804	8	313
tr23	5832	6	204
tr31	10128	7	927
tr41	7454	10	878
tr45	8261	10	690

For 20NG, we created three sets of groups, as shown in Table 2. As in [6,16], 50 documents were sampled from each group in order to create one dataset, and 10 datasets were created for each set of groups. For each dataset, we conducted stemming using porter stemmer[5] and MontyTagger[6], removed stop words, and selected 2,000 words with large mutual information [5]. For TREC datasets, we utilized 6 datasets as in [16]. Their characteristics are summarized in Table 1.

Evaluation Measures. For each dataset, the cluster assignment was evaluated w.r.t. Normalized Mutual Information (NMI) [15,16]. Let T, $\hat{T}$ stand for the random variables over the true and assigned clusters. NMI is defined as

$$NMI = \frac{I(\hat{T};T)}{(H(\hat{T}) + H(T))/2} \quad (\in [0,1]) \tag{16}$$

[2] skmeans is a standard clustering algorithm for high-dimensional sparse data.
[3] http://people.csail.mit.edu/ jrennie/20Newsgroups/. 20news-18828 was utilized.
[4] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download
[5] http://www.tartarus.org/∼martin/PorterStemmer
[6] http://web.media.mit.edu/∼hugo/montytagger

Table 2. Datasets from 20 Newsgroup dataset

dataset	included groups
Multi5	comp.graphics, rec.motorcycles,rec.sport.baseball, sci.space talk.politics.mideast
Multi10	alt.atheism, comp.sys.mac.hardware,misc.forsale, rec.autos,rec.sport.hockey, sci.crypt,sci.med, sci.electronics,sci.space,talk.politics.guns
Multi15	alt.atheism, comp.graphics, comp.sys.mac.hardware, misc.forsale, rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, talk.politics.guns, talk.politics.mideast, talk.politics.misc

where $H(T)$ is Shannon Entropy. NMI corresponds to the accuracy of assignment. The larger NMI is, the better the result is.

All the compared methods first construct the representation for clustering and then apply some clustering method (e.g., skmeans). Thus, the running time (CPU time in second) for the first part (i.e., representation construction) was measured on a computer with Windows Vista, Intel Core2 Quad Q8200 2.33 GHz, 2G memory. All the methods were implemented with R and R packages.

Comparison. We compared our approach with SCREEN [16] and PCP [13]. Since all the compared methods are partitioning based clustering methods, we assume that the number of clusters k is specified.

SCREEN [16] conducts semi-supervised clustering by projecting the feature representation onto the subspace where the covariance is maximized. The covariance matrix w.r.t. the original representation is constructed and their eigenvectors are utilized for projection. For high-dimensional data such as documents, this process is rather time consuming, since the number of attributes (e.g., terms) gets large. Thus, PCA (Principal Component Analysis) was first utilized as preprocessing to reduce the number of dimension, and SCREEN was applied to the pre-processed data in [16]. We followed this process in the evaluation.

PCP [13] first conducts metric learning based on the semi-definite programming, and then kernel k-means clustering is conducted over the learned metric. Some package (e.g. Csdp) is utilized to solve the semi-definite programming based on the specified pairwise constraints and similarities.

Parameters. The parameters under the pairwise constraints in Definition 1 are: 1) the number of constraints, and 2) the pairs of instances for constraints. As for 2), pairs of instances were randomly sampled from each dataset to generate the constraints. Thus, the main parameter is 1), the number of constraints, for $\boldsymbol{C}_{ML}$ and $\boldsymbol{C}_{CL}$. We set $|C_{ML}| = |C_{CL}|$, and varied the number of constraints.

Each data $\boldsymbol{x}$ in the dataset was normalized such that $\boldsymbol{x}^t\boldsymbol{x} = 1$, and Euclidian distance was utilized for SCREEN as in [16]. With this normalization, cosine similarity, which is widely utilized as the standard similarity measure in document processing, was utilized for GBSSC and PCP, and the initial edge-weighted graph for each dataset was constructed with the similarities. The dimension l of the subspace was set to the number of clusters k. In addition, following the

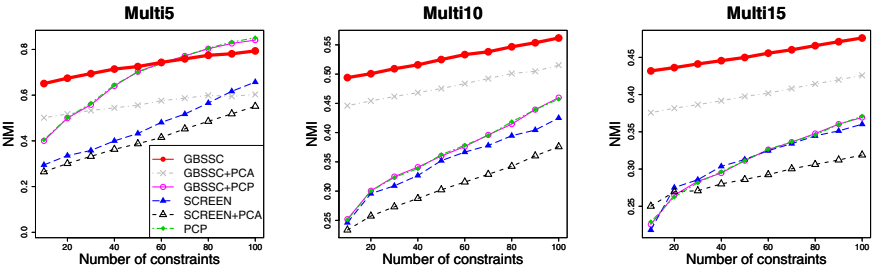


Fig. 3. Result on 20-Newsgroup (NMI)

procedure in [13], m-nearest neighbor graph was constructed for PCP with $m = 10$. λ in eq.(7) was set to 0.5.

Evaluation Procedure. For each number of constraints, the pairwise constraints $\boldsymbol{C}_{ML}$ and $\boldsymbol{C}_{CL}$ were generated randomly based on the ground-truth label in the datasets, and clustering was conducted with the generated constraints. Clustering with the same number of constraints was repeated 10 times with different initial configuration in clustering. In addition, the above process was also repeated 10 times for each number of constraints. Thus, for each dataset and the number of constraints, 100 runs were conducted and their average is reported in the following section.

3.2 Evaluation on Real World Datasets

This section reports the results for the datasets in Section 3.1. In the reported figures, horizontal axis corresponds to the number of constraints; vertical one corresponds to either NMI in eq.(16) or CPU time (in sec.). In the legend in the figures, red lines correspond to the proposed GBSSC, blue to SCREEN, and green to PCP. Also, +PCA stands for the case where the dataset was first pre-processed by PCA (using 100 eigenvectors as in [16]) and then the corresponding method was applied. GBSSC+PCP corresponds to the situation where must-links were handled by the proposed contraction in Section 2.4 and cannot-links by PCP.

20 Newsgroup Datasets. The results for 20NG datasets in Table 2 are summarized in Figs. 3(NMI) and 4 (CPU time). These are the average of 10 datasets for each set of groups (i.e., average of 1000 runs). The results indicate that the proposed GBSSC outperformed other methods w.r.t. NMI (Fig. 3) when l=k[7]. For Multi5, although the performance of PCP got close to that of GBSSC as the number of constraints increased, Fig.4 shows that GBSSC was faster more than two orders of magnitude (100 times faster). Likewise, GBSSC+PCP and PCP was almost the same w.r.t. NMI, but the former was faster with more than one order (10 times faster).

[7] The dimension l of the subspace is equal to the number of clusters k. Note that we did not conduct any tuning for the value of l in these experiments.

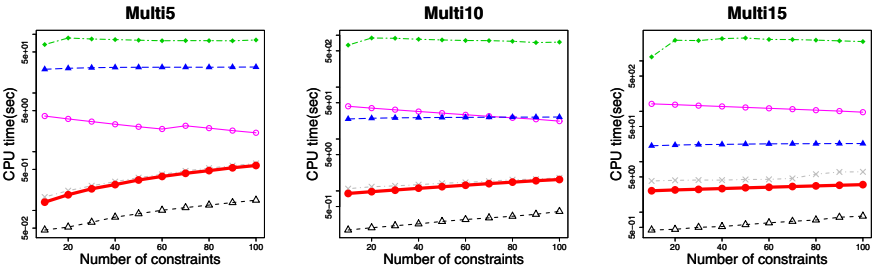


Fig. 4. Result on 20-Newsgroup (CPU time)

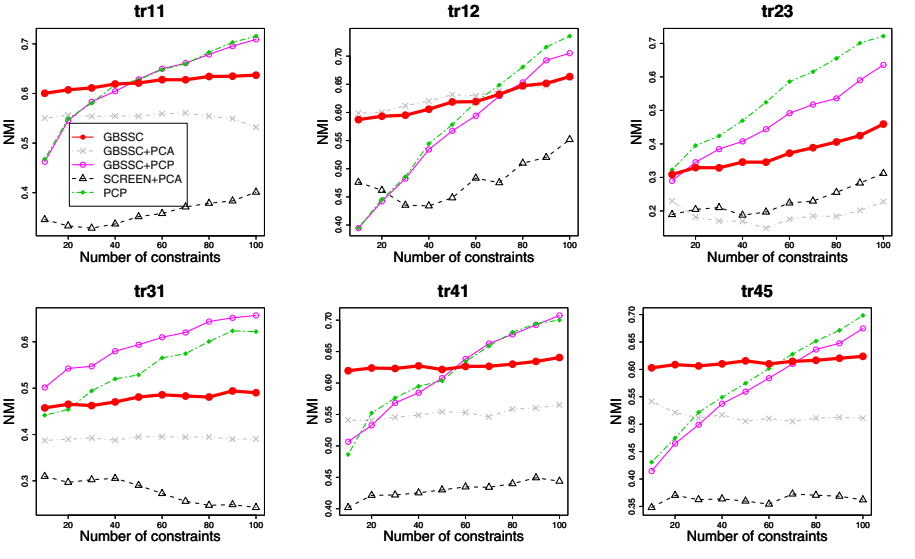


Fig. 5. Result on TREC dataset(NMI)

Dimensionality reduction with PCA was effective for the speed-up of SCREEN, but it was not for GBSSC (Fig.4). On the other hand, it *deteriorated* their performance w.r.t. NMI. Thus, it is not necessary to utilize pre-processing such as PCA for the proposed GBSSC, and still GBSSC showed better performance.

TREC Datasets. The results for TREC datasets are summarized in Figs. 5 and 6. As in 20NG, GBSSC outperformed SCREEN w.r.t. NMI[8]. Also, it outperformed PCP for tr12 and tr45. On the other hand, GBSSC outperformed PCP when the number of constraints were small for tr11 and tr41, but the latter was better for tr23 and tr31. However, as in 20NG, GBSSC was faster than PCP more than two orders of magnitude.

[8] Without using PCA, SCREEN took much time and thus is not reported for TREC.

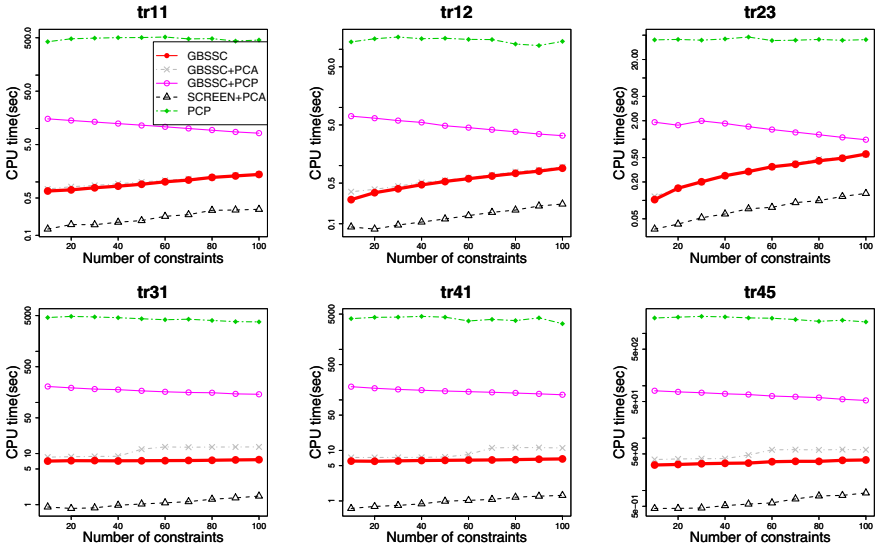


Fig. 6. Results on TREC datasets (CPU time)

3.3 Discussions

The results show that the proposed approach is effective in terms of the accuracy (NMI) and the running time (CPU time). GBSSC outperformed SCREEN in both NMI and CPU time. Although it did not outperformed PCP in some TREC datasets w.r.t. NMI, but it was faster more than two orders of magnitude. From these results, the proposed method is effective in terms of the balance between the accuracy of cluster assignment and running time. Especially, it could leverage small amount of pairwise constraints to increase the performance. We believe that this is a good property in the semi-supervised learning setting.

In our approach graph contraction is utilized, not for reducing the number of instances, but for enforcing the must-link constraint. In addition, in order to reflects cannot-link constraints, pairwise relations (constraints and similarities) among instances are utilized in the proposed objective function in eq.(15). Although the proposed approach utilizes graph Laplacian as in [2], it differs since pairwise constraints for semi-supervised clustering are also utilized for constructing the projected representation.

4 Concluding Remarks

This paper proposed a graph-based projection approach for semi-supervised clustering based on pairwise relations among instances. The entire data are represented as an edge-weighted graph with the pairwise similarities among instances. The graph structure enables to deal with two kinds of constraints as well as

pairwise similarities over the same unified representation. In order to reflect the constraints on the clustering process, the constructed graph is modified by contraction in graph theory and graph Laplacian in spectral graph theory. By exploiting the constraints as well as similarities among instances, the entire data are projected onto a subspace via the modified graph, and clustering is conducted over the projected representation.

The proposed approach was evaluated over several real world datasets. The results indicate that it is effective in terms of the balance between the accuracy of cluster assignment and running time. We plan to evaluate the proposed method with other real world datasets such as image datasets, and to improve our approach based on the obtained results.

Acknowledgments

This work is partially supported by the grant-in-aid for scientific research (No. 20500123) funded by MEXT, Japan.

References

1. Basu, S., Bilenko, M., Mooney, R.J.: A probabilistic framework for semi-supervised clustering. In: KDD 2004 (2004)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15, 1373–1396 (2002)
3. Chapelle, O., Schölkopf, B., Zien, A. (eds.): Semi-Supervised Learning. MIT Press, Cambridge (2006)
4. Chung, F.: Spectral Graph Theory. American Mathematical Society, Providence (1997)
5. Cover, T., Thomas, J.: Elements of Information Theory. Wiley, Chichester (2006)
6. Dhillon, J., Mallela, S., Modha, D.: Information-theoretic co-clustering. In: Proc. of KDD 2003, pp. 89–98 (2003)
7. Dhillon, J., Modha, D.: Concept decompositions for lage sparse text data using clustering. Machine Learning 42, 143–175 (2001)
8. Diestel, R.: Graph Theory. Springer, Heidelberg (2006)
9. Elghazel, H., Yoshida, T., Deslandres, V., Hacid, M., Dussauchoy, A.: A new greedy algorithm for improving b-coloring clustering. In: Escolano, F., Vento, M. (eds.) GbRPR. LNCS, vol. 4538, pp. 228–239. Springer, Heidelberg (2007)
10. Guënoche, A., Hansen, P., Jaumard, B.: Efficient algorithms for divisive hierarchical clustering with the diameter criterion. J. of Classification 8, 5–30 (1991)
11. Hartigan, J., Wong, M.: Algorithm AS136: A k-means clustering algorithm. Journal of Applied Statistics 28, 100–108 (1979)
12. Jain, A., Murty, M., Flynn, P.J.: Data clustering: A review. ACM Computing Surveys 31, 264–323 (1999)
13. Li, Z., Liu, J., Tang, X.: Pairwise constraint propagation by semidefinite programming for semi-supervised classification. In: ICML 2008, pp. 576–583 (2008)
14. Ng, R., Han, J.: Clarans: a method for clustering objects for spatial data mining. IEEE Transactions on Knowledge and Data Engineering 14(5), 1003–1016 (2002)

15. Strehl, A., Ghosh, J.: Cluster Ensembles -A Knowledge Reuse Framework for Combining Multiple Partitions. J. Machine Learning Research 3(3), 583–617 (2002)
16. Tang, W., Xiong, H., Zhong, S., Wu, J.: Enhancing semi-supervised clustering: A feature projection perspective. In: Proc. of KDD 2007, pp. 707–716 (2007)
17. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
18. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: ICML 2001, pp. 577–584 (2001)
19. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side-information. In: NIPS, vol. 15, pp. 505–512 (2003)

Self-organisation in an Agent Network via Multiagent Q-Learning

Dayong Ye[1], Minjie Zhang[1], Quan Bai[2], and Takayuki Ito[3]

[1] University of Wollongong, Wollongong, NSW 2522 Australia
[2] CSIRO ICT Centre, Hobart, TAS 7001 Australia
[3] Nagoya Institute of Technology, Nagoya 466-8555 Japan
{dy721,minjie}@uow.edu.au, Quan.Bai@csiro.au, ito@nitech.ac.jp

Abstract. In this paper, a decentralised self-organisation mechanism in an agent network is proposed. The aim of this mechanism is to achieve efficient task allocation in the agent network via dynamically altering the structural relations among agents, i.e. changing the underlying network structure. The mechanism enables agents in the network to reason with whom to adapt relations and to learn how to adapt relations by using only local information. The local information is accumulated from agents' historical interactions with others. The proposed mechanism is evaluated through a comparison with a centralised allocation method and the *K-Adapt* method. Experimental results demonstrate the decent performance of the proposed mechanism in terms of several evaluation criteria.

1 Background

To cope with complex tasks, agents are usually organised in a network where an agent interacts only with its immediate neighbours in the network. Self-organisation, defined by Serugendo et al. [1] as "*the mechanism or the process enabling the system to change its organisation without explicit external command during its execution time*", can be used in agent networks to improve the cooperative behaviours of agents. Mathieu et al. [2] provided three principles for self-organisation agent networks design, including (1) creation of new specific relations between agents in order to remove the middle-agents, (2) exchange of skills between agents to increase autonomy, and (3) creation of new agents to reduce overloading. In this paper, our contribution focuses on the first principle provided by Mathieu et al., i.e. modification of existing relations between agents to achieve a better allocation of tasks in distributed environments.

Currently, research on self-organisation mechanisms in multiagent and agent-based complex systems has produced results of significance. Horling et al. [3] proposed an organisation structure adaptation method, which employed a central blackboard, by using self-diagnosis. Their method involves a diagnostic subsystem for detecting faults in the organisation. Then, against these faults, some fixed pre-designed reorganisation steps are launched. Hoogendoorn [4] presented an approach based on max flow networks to dynamically adapt organisational

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 14–26, 2010.

models to environmental fluctuation. His approach assumed two special nodes in the network, namely the *source node* with indegree of 0 and the *sink node* with outdegree of 0. These two special nodes make the approach centralised, since if these two nodes are out of order the organisation adaptation process might be impacted. Hence, the approaches proposed by both Horling et al. [3] and Hoogendoorn [4] are centralised in nature and have the potential of single point failure. Kamboj and Decker [5] developed a self-organisation method based on organisational self-design, which achieved self-organisation by dynamically generating and merging agents in response to the changing requirements. Nevertheless, the internal framework of agents might not be changed on all occasions because of physical and accessibility limitations (e.g. a remote central server, which is modeled as an agent, cannot easily be replicated).

Self-organisation methods focusing on network structural adaptation (namely modifying relations among agents), which are used to improve team formation or task allocation, have also been researched. Gaston and desJardins [6] developed two network structural adaptation strategies for dynamic team formation, i.e. structure-based approach and performance-based approach. The two strategies are suitable in different situations. Glinton et al. [7] empirically analysed the drawback of the structure-based strategy proposed by Gaston and desJardins [6], and then designed a new network adaptation strategy which limits the maximum number of links an agent can have. Abdallah and Lesser [8] did further research in self-organisation of agent networks and creatively used reinforcement learning to adapt the network structure by allowing agents to not only adapt the underlying network structure during the learning process but also use information from learning to guide the adaptation process. The common limitation of these works is that they assumed that only one type of relations exists in the network and the number of neighbours possessed by an agent has no effect on its local load. These assumptions are impractical in some cases where multiple relations exist among agents in a network and agents have to expend resources to manage their relations with other agents. To overcome this common limitation, Kota et al. [9] [10] devised a network structural adaptation mechanism through meta-reasoning. Their mechanism primarily supposed that multiple relations exist in the network and considered the load of agents to manage their relations. Nonetheless, their mechanism is somewhat biased from the agent which initialises the relation adaptation process towards the agent which is requested to accessorily adapt relation, because the initiative agent evaluates most attributes regarding relation adaptation. This bias might cause the initiative agents a little subjective when making decisions.

Against this background, in this paper, we propose a self-organisation mechanism, called *Learn-Adapt*. This mechanism adopts a token based approach, and can achieve decentralised structural adaptations via multiagent Q-learning. Xu et al. [11] have shown that token-based mechanisms can collect as much information as broadcast mechanisms while using much less bandwidth. Compared with current approaches, our mechanism utilises only local information to adapt the network structure and considers multiple relations in the network,

communication efficiency of different relations, the management cost effect on agents which is brought by their neighbours, and the potential benefit after changing relations. According to our mechanism, when an agent intends to change their relation with another agent, the two agents independently evaluate their rewards about changing relations through learning the Q-value of each available action and then the two agents jointly make a reasonable and optimal decision about changing relations. In contrast to the *K-Adapt* mechanism, proposed by Kota et al. [10], our *Learn-Adapt* mechanism is unbiased for both agents which jointly adapt their relation, and we empirically demonstrate that the performance of our mechanism, *Learn-Adapt*, is better than *K-Adapt* in different situations. The rest of the paper is organised as follows. Section 2 introduces our agent network model for task allocation. Section 3 illustrates the network performance evaluation approach and proposes a decentralised network structural adaptation mechanism. Experimental results and analysis are presented in Section 4. The paper is concluded in Section 5.

2 The Agent Network Model

The aim of the agent network is to assign tasks to agents such that the communication cost among agents is minimised and the benefit obtained by completing tasks is maximised. In our model, an agent network comprises a set of collaborative agents, i.e. $A = \{a_1, ..., a_n\}$, situated in a distributed task allocation environment. The task environment presents a continuous dynamic stream of tasks that have to be performed. Each task, Θ, is composed of a set of subtasks, i.e. $\Theta = \{\theta_1, ..., \theta_m\}$. Each subtask, $\theta_i \in \Theta$, requires a particular resource and a specific amount of computation capacity to fulfill. In addition, each subtask has a relevant benefit paid to the agent which successfully completes the subtask. A subtask θ_i is modeled as a token Δ_i which can be passed in the network to find a suitable agent to complete. Each token consists of not only the information about resource and computation requirement of the corresponding subtask, but also the token traveling path which is composed of those agents that the token has passed.

In the agent network, instead of a single type of neighbours, there are three types of neighbours, namely *peer*, *subordinate* and *superior* neighbours, which are constituted by two relations, i.e. *peer-to-peer* and *subordinate-superior* relations. The formal definitions of these two relations are given below.

Definition 1. (*Peer-to-Peer*). A *peer-to-peer relation*, denoted as "$\sim$" ($\sim \subseteq A \times A$), is a *Compatible Relation*, which is reflexive and symmetric, such that $\forall a_i \in A : a_i \sim a_i$ and $\forall a_i, a_j \in A : a_i \sim a_j \Rightarrow a_j \sim a_i$.

Definition 2. (*Subordinate-Superior*). A *subordinate-superior relation*, written as "$\prec$" ($\prec \subseteq A \times A$), is a *Strict Partial Order Relation*, which is irreflexive, asymmetric and transitive, such that $\forall a_i \in A : \neg(a_i \prec a_i)$, $\forall a_i, a_j \in A : a_i \prec a_j \Rightarrow \neg(a_j \prec a_i)$ and $\forall a_i, a_j, a_k \in A : a_i \prec a_j \wedge a_j \prec a_k \Rightarrow a_i \prec a_k$.

According to the above definitions, there are three neighbour types, which are *peer*, *subordinate* and *superior* neighbours. For convenience, we stipulate that relation $\succ$ is the reverse relation of $\prec$, namely $a_i \prec a_j \Leftrightarrow a_j \succ a_i$. In this paper, it is assumed that there is at most one relation between two agents in the network. Figure 1 displays an example agent network. The contents in parenthesis denote the resources supplied by an agent.

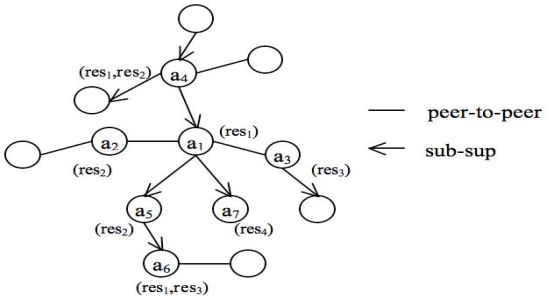


Fig. 1. An Example Agent Network at Time t

Each agent is of the form $a_i = \langle Res_i, Comp_i \rangle$, where Res_i is the set of resources possessed by a_i and $Comp_i$ is the computation capacity of a_i for completing tasks. Moreover, the knowledge of each agent is modeled as a tuple $\langle Neig_i(t), Act_i(t), Tokens_i(t) \rangle$. The first element, $Neig_i(t)$, is the neighbours set of agent a_i at time t. $Neig_i(t)$ can be further divided into three subsets, $Neig_i^{\sim}(t)$, $Neig_i^{\prec}(t)$ and $Neig_i^{\succ}(t)$. $Neig_i^{\sim}(t)$ contains the *peer*s of a_i, $Neig_i^{\prec}(t)$ is consisted of the direct *superior*s of a_i, and $Neig_i^{\succ}(t)$ comprises of the direct subordinates of a_i.

The second element in agent knowledge tuple, $Act_i(t)$, is the available action set of agent a_i at time t. Before description of the available action set, we first introduce the definition of action.

Definition 3. (*Action*) An *action* is defined as a decision made by an agent to change relation with another agent.

There are seven different atomic actions defined in our model, which are $form\_\sim$, $form\_\prec$, $form\_\succ$, $dissolve\_\sim$, $dissolve\_\prec$, $dissolve\_\succ$ and *no_action*. For example, if agent a_i performs action $form\_\prec$ with agent a_j, a_i will become a *subordinate* of a_j. Obviously, actions $form\_\prec$ and $form\_\succ$ are the reverse action of each other, namely that if a_i performs $form\_\prec$ with a_j, a_j has to take $form\_\succ$ with a_i. The atomic actions can be combined together. The meanings of combination actions can be easily deduced from the meanings of atomic actions. For example, the combination action, $dissolve\_\prec + form\_\sim$, which is taken by a_i with a_j, implies that a_i first dissolves a_j from a_i's *superior* and forms a *peer* relation with a_j. It should be noted that an agent at different time steps might possess different available actions.

The possible choices of actions available to agents in different situations are illustrated as follows.

1. There is no relation between agents a_i and a_j. The possible choices of actions include $form\_\sim$, $form\_\prec$, $form\_\succ$ and *no_action*.

2. a_i is a *peer* of a_j, i.e. $a_i \sim a_j$. The possible actions involve $dissolve\_\sim$, $dissolve\_\sim + form\_\prec$, $dissolve\_\sim + form\_\succ$ and *no_action*.

3. a_i is a *subordinate* of a_j, i.e. $a_i \prec a_j$. The possible actions include $dissolve\_\prec$, $dissolve\_\prec + form\_\sim$, $dissolve\_\prec + form\_\succ$ and *no_action*. These actions are based on a_i's perspective, while, on a_j's view, a_j needs to reverse these actions.

4. a_i is a *superior* of a_j, i.e. $a_j \prec a_i$. This situation is the reverse condition of $a_i \prec a_j$.

Definition 4. (*Action Set*). An *action set*, $Act_i(t)$, is defined as a set of available actions for the agent a_i at time t, which includes some atomic actions or combination actions.

The last element in the agent knowledge tuple, $Tokens_i(t)$, stores not only the tokens agent a_i currently holds at time t but also all the previous tokens incoming and outgoing through a_i. In this paper, we assume that each agent has an infinite amount of memory to store tokens.

Furthermore, an agent possesses information about the resources it provides, the resources its *peer*s could provide, and the resources all of its *subordinate*s and its direct *superior* could provide, although the agent might have no idea exactly which *subordinate* owns which resource.

During the allocation of a subtask θ, an agent a_i always tries to execute the subtask by itself if it has adequate resources and computation capacity. Otherwise, a_i will generate a token for the subtask and pass the token to one of its *subordinate*s which contains the expected resource. Since a_i does not know which *subordinate* has which *resource*, the token might be passed several steps in the agent network forming a delegation chain. If a_i finds no suitable subordinate (no subordinate contains the expected resource), it will try to pass the token to its *peer*s. In the case that no peer is capable of the subtask, a_i will pass the token back to one of its *superior*s which will attempt to find some other subordinates or peers for delegation.

Apparently, the structure of the agent network will influence the task allocation process. In the next section, we will describe the self-organisation mechanism to adapt the structure of the agent network, involving an evaluation method to measure the profit of the network.

3 Decentralised Self-organisation Mechanism

Before devising a decentralised self-organisation mechanism, it is necessary to introduce an evaluation method to estimate the profit of the agent network. Towards this goal, we illustrate the concept of evaluation criteria, which includes cost, benefit, profit and reward of an agent and, further, the agent network.

3.1 Network Performance Evaluation

The cost, benefit and profit of the network are calculated after a predefined number of tasks are allocated, each of which contains several subtasks. The cost of the agent network, $Cost_{NET}$, consists of four attributes, i.e. communication cost, computation cost consumed by agents to complete assigned subtasks, management cost for maintaining subtasks and management cost for keeping neighbour relations with other agents. Due to the page limitation, the detailed calculation of each attribute of $Cost_{Net}$ cannot be presented.

The benefit of the network, $Benefit_{NET}$, is simply the sum of benefits obtained by all the agents in the network. The benefit of each agent depends on how many subtasks are completed by that agent. As depicted in Section 2, each task Θ contains several subtasks, $\theta_1, \theta_2, ...$, represented as tokens Δ_1, Δ_2.... When a subtask θ_i is successfully completed, the agent which executes this subtask can obtain the relevant benefit.

Finally, the profit of the entire network, $Profit_{NET}$, is:

$$Profit_{NET} = Benefit_{NET} - Cost_{NET} \tag{1}$$

3.2 Self-organisation Mechanism Design

The aim of our self-organisation mechanism is to improve the profit of the agent network during task allocation processes via changing the network structure, i.e. changing the relations among agents. Our mechanism is based on the historical information of individual agents. Specifically, agents use the information about the former task allocation processes to evaluate their relations with other agents. We formulate our self-organisation mechanism by using a multiagent Q-learning approach. The reason for choosing the Q-learning approach is that it provides a simple and suitable methodology for representing our mechanism in terms of actions and rewards. Before describing our self-organisation mechanism, we first consider a simple scenario with two agents, a_i and a_j, and three available actions for each agent. The reward matrix of the two agents is displayed in Table 1.

Each cell $(r_i^{x,y}, r_j^{x,y})$ in Table 1 represents the reward received by the row agent (a_i) and the column agent (a_j), respectively, if the row agent a_i plays action x and the column agent a_j plays action y.

The reward of each agent, r_i, is based on how much load could be reduced on agent a_i and how much load could be decreased on the intermediate agents,

Table 1. Reward Matrix of a_i and a_j

a_i \ a_j	$form\_\prec$	$form\_\sim$	$form\_\succ$
$form\_\succ$	$r_i^{1,1}, r_j^{1,1}$	$r_i^{1,2}, r_j^{1,2}$	$r_i^{1,3}, r_j^{1,3}$
$form\_\sim$	$r_i^{2,1}, r_j^{2,1}$	$r_i^{2,2}, r_j^{2,2}$	$r_i^{2,3}, r_j^{2,3}$
$form\_\prec$	$r_i^{3,1}, r_j^{3,1}$	$r_i^{3,2}, r_j^{3,2}$	$r_i^{3,3}, r_j^{3,3}$

and how much potential benefit might be obtained by agent a_i in the future. Here, an intermediate agent is an agent which resides on a token path, written as $\Delta.path$. For example, agent a_i has no relation with agent a_j, but a_i received many subtasks from a_j during former task allocation processes. a_i, then, makes the decision with regard to forming a relation with a_j. If a_i would like to form the *subordinate-superior* relation with a_j, i.e. performing the action $form\_\prec$ (for a_j, performing the reverse action $form\_\succ$), the management cost on both a_i and a_j will rise because both a_i and a_j have to maintain a new neighbour. Nevertheless, those agents, which are in the $\Delta.path$, could save communication cost (which are considered as a_i's reward when a_i makes decisions), since they do not need to pass tokens between agents a_i and a_j any more. Here, Δ refers to the tokens that are held by a_i and sent by a_j. For a_j, it could save management cost for maintaining subtasks, as a_j can directly pass tokens to a_i without waiting for intermediate agents to pass tokens and, hence, a_j's subtasks could be allocated in less time steps. The potential benefit which would be obtained by a_i is evaluated on the basis of the benefit a_i gained for completing the subtasks assigned by a_j, while the potential benefit of a_j is calculated in an analytical way. We suppose that the action $form\_\prec$, with a_i as the *subordinate* and a_j as the *superior*, can make a_i potentially receive more subtasks from a_j and then get more benefits. *Algorithm 1* demonstrates our self-organisation mechanism in pseudocode form.

Algorithm 1. Reorg. Mechanism according to a_i

```
1  Candidates_i ← a_i selects agents in the network;
2  for each a_j ∈ Candidates do
3      Act_i ← available_actions(a_i, a_j);
4      Act_j ← available_actions(a_i, a_j);
5      for each x ∈ Act_i, y ∈ Act_j do
6          Initialise Q_ix and Q_jy arbitrarily;
7          for k = 0 to a predefined integer do
8              calculate π_ix(k) and π_jy(k);
9              Q_ix(k+1) = Q_ix(k)+
                 π_ix(k)α(Σ_y r_i^{x,y} π_jy(k) − Q_ix(k));
10             Q_jy(k+1) = Q_jy(k)+
                 π_jy(k)α(Σ_x r_j^{x,y} π_ix(k) − Q_ix(k));
11         end for
12     end for
13     ⟨x_opti, y_opti⟩ ← argMax_{match(x,y)}(Q_ix + Q_jy);
14     a_i, a_j take actions x_opti and y_opti, respectively;
15     end if
16 end for
```

The first component (Line 1) refers to the reasoning aspect, which is displayed in *Algorithm 2*, about selecting agents to initiate the self-organisation process. After selection, both agents, a_i and a_j, estimate which actions are available at the current state (Lines 3 and 4) as described in Section 2. Then, a_i and a_j learn the Q-value of each available action, separately (Lines 5-11). In Line 6,

the Q-value of each action is initialised arbitrarily. In Line 8, π_{ix} indicates the probability regarding agent a_i taking the action x. To calculate π_{ix}, we employ the ϵ-greedy exploration method devised by Gomes and Kowalczyk [12] shown in Equation 2, where $0 < \epsilon < 1$ is a small positive number and n is the number of available actions possessed by agent a_i.

$$\pi_{ix} = \begin{cases} (1-\epsilon) + (\epsilon/n), \text{if } Q_{ix} \text{ is the highest} \\ \epsilon/n \text{ , otherwise} \end{cases} \tag{2}$$

Furthermore, in Line 9, Q_{ix} is the Q-value of action x taken by agent a_i. In Lines 9 and 10, $0 < \alpha < 1$ is the learning rate.

When finishing learning Q-values, a_i and a_j (Line 13) cooperate to find the optimal actions for both of them, where $match(x, y)$ is a function which is used to test whether the actions x and y that are taken by a_i and a_j, respectively, are matched. An action is only matched with its reverse action, such as that $form\_\prec$ is only matched with $form\_\succ$. Therefore, a_i and a_j have to cooperate to find the actions, which can be matched together and make the sum of their Q-values maximum.

Algorithm 2. Candidates selection of each agent

```
1  for each a_i ∈ A do
2      Candidates_i ← ∅;
3      for each Δ_k ∈ Tokens_i do
4          statistics of Δ_k.owner;
5      end for
6      if ∃ # of same Δ_k.owner > thre_1 and
7          Δ_k.owner ∉ Neig_i^≁ ∨ Neig_i^≻ ∨ Neig_i^≺ then
8          Candidates_i ← Candidates_i ∪ {Δ_k.owner};
9      end if
10     if ∃ # of same Δ_k.owner < thre_2 and
11         Δ_k.owner ∈ Neig_i^≁ ∨ Neig_i^≻ ∨ Neig_i^≺ then
12         Candidates_i ← Candidates_i ∪ {Δ_k.owner};
13     end if
14 end for
```

Algorithm 2 illustrates the reasoning aspect of each agent for selecting a group of agents to initilise the self-organisation process. As described in Section 2, each agent has not only the tokens it currently holds but also all the previous tokens incoming and outgoing through it. Then, each agent uses the local information provided by the tokens to choose candidates. Firstly, from Lines 3 to 5, agent a_i identifies the owner of each token stored in a_i's token list, $Tokens_i$, and counts the number of tokens from each owner. On the one hand, if the number of tokens from one owner exceeds a predefined threshold and this owner is not a neighbour of a_i, this owner will be added into the candidates set (Lines 6-9). This can be explained that if an agent is not a neighbour of agent a_i but often delegated tasks to a_i in the former task allocation processes, a_i might want to adapt the

relation with this agent. On the other hand, if the number of tokens from one owner is lower than another predefined threshold and this owner is a neighbour of a_i, this owner will be also appended into the candidates set (Lines 10-13). This can be explained that if an agent, which is a neighbour of a_i, delegated very few tasks to a_i in previous task allocation processes, then a_i might also want to alter the relation with this agent.

According to the description, our self-orgnisation mechanism in an agent network is based on only local information which is represented by tokens. In addition, both agents, a_i and a_j, employ the multiagent Q-learning method to learn optimal actions. In this way, our mechanism enables every pair of agents to independently learn the Q-value for taking any of the available actions towards altering their relation, and the two agents jointly select the actions which can be matched together and maximise the sum of their Q-values. In this manner, our mechanism could make the pair of agents be treated fairly, when they decide to change their relation.

4 Experiment and Analysis

In this section, the effectiveness of our self-organisation mechanism is demonstrated through experimental evaluation. We first describe the experimental setup and thereafter present the experimental results and analysis.

4.1 Experimental Setup

To objectively exhibit the effectiveness of our self-organisation mechanism, *Learn-Adapt*, we compare our mechanism with two other methods, namely *Central* and *K-Adapt* [10], which are depicted as follows.

1. *Central*: This is an ideal centralised task allocation mechanism in which there is an external omniscient central manager that maintains information about all the agents and tasks in the network. The central manager is able to interact with all the agents in the network without cost. Thereby, all task allocations are only one step direct allocation through the central manager. This method is not practical or robust, but it can be used as an upper bound of the performance of an organisation in our experiment.
2. *K-Adapt*: This method was proposed by Kota et al. [10], which utilised meta-reasoning approach to adapt the relation between two agents. This mechanism is somewhat biased from the agent which launches the relation adaptation process towards the agent which is requested to accessorily adaptation relation.

In this experiment, the agent organised network is generated by using the Small World network [13], in which most neighbours of an agent are connected to each other. Nevertheless, the approach presented by [13] deals with only one relation between agents in the Small World network. We, thus, modify the approach to accommodate multiple relations by randomly changing the relation between two neighbouring agents. Moreover, in order to control the number of resources an agent could hold, a parameter called *Resource Probability* (RP) is utilised, such

that an agent is assigned a resource with probability RP. Hence, with the increase of RP, agents could possess more resources. For simplicity, tasks are created by randomly generating resource and computation capacity requirements, and each task is randomly distributed to one of the agents in the network. Finally, the evaluated criteria is $Profit_{NET}$ (Equation 1), obtained by both *K-Adapt* and our *Learn-Adapt* in a percentage format with the maximum network profit gained by the *central* mechanism. For clarity, the values of parameters which are exploited in this experiment and their meanings are listed in Table 2.

Table 2. Parameters Setting

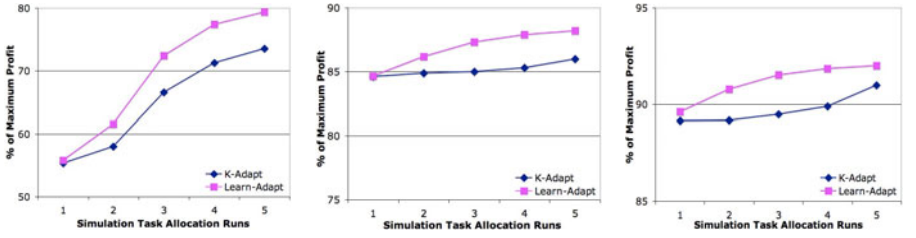
Parameters	Values	Explanations
n	$50 \sim 250$	The number of agents
deg	$4 \sim 10$	The average number of neighbours
RP	$0.1 \sim 0.6$	Resource Probability
m	15000	The number of tasks
α	0.2	Learning rate
ϵ	0.4	Action selection distribution probability
k	100	Learning rounds
$thre_1$, $thre_2$	2, 5	Thresholds for choosing agents to adapt

4.2 Experimental Results and Analysis

Figure 2 demonstrates the percentage profits obtained by both *K-Adapt* and *Learn-Adapt* with different resource probabilities (RP), compared with the maximum profit which is gained by *Central*. The number of agents in the network, n, is fixed at 50 and the average number of neighbours of each agent, deg, is set to 6. The x-aix represents the number of simulation task allocation runs and each run consists of 15000 tasks. It can be seen that *Learn-Adapt* performs consistently better than *K-Adapt* in all situations. In Figure 2(a) ($RP = 0.1$), with more task allocation runs, the difference between *Learn-Adapt* and *K-Adapt* is gradually increasing. This is because when each agent has very few resources, agents have to allocate tasks to others. Thus, an effective network structure could reduce agents' communication cost and management cost, and, further, raise the profit of the entire network. In this case, a smarter mechanism could bring better performance through generating a more effective network structure.

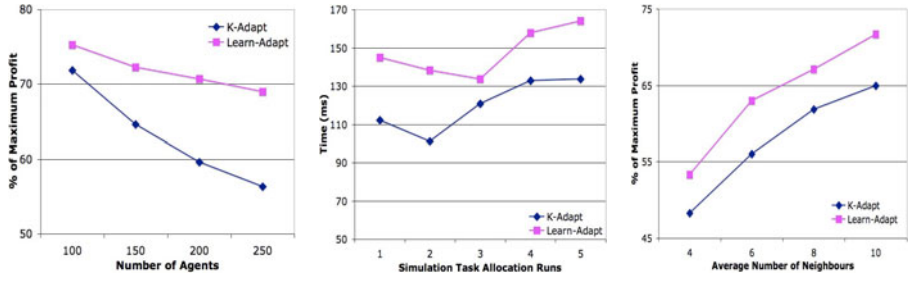
With the increase of resource probability (Figures 2(b) and 2(c)), both *K-Adapt* and *Learn-Adapt* could achieve better performance. This can be explained that with higher resource probability, each agent would have more resources and, thus, could fulfill more tasks by itself. It should also be noted that the difference between *Learn-Adapt* and *K-Adapt* narrows as the resource probability rises. This is because when agents become more homogeneous, a smarter method cannot correspondingly bring more profit for the network.

Furthermore, we estimated both *Learn-Adapt* and *K-Adapt* in other three aspects as well, which include their performance in different network scales ($n =$



(a) Resource Probability (RP)=0.1 (b) Resource Probability (RP)=0.3 (c) Resource Probability (RP)=0.6

Fig. 2. Relative Profits of *K-Adapt* and *Learn-Adapt* with Different Resource Probabilities (RP)



(a) Different Number of Agents (b) Time Consumption (c) Different Number of Neighbours

Fig. 3. Other Performance Indices of *K-Adapt* and *Learn-Adapt*

$100 \sim 250, deg = 6, RP = 0.1$), their overhead ($n = 250, deg = 6, RP = 0.1$) and their adaptability ($n = 250, deg = 4 \sim 10, RP = 0.1$). Figures 3(a), 3(b) and 3(c) show the experimental results of the three aspects, respectively. In Figure 3(a), it can be found that with the network scale increasing, i.e. more agents forming the network, the performance of both *Learn-Adapt* and *K-Adapt* decreases. This is because that, with the increase of network scale, each agent has to form a longer delegation chain to allocate its tasks, which will incur more communication cost and management load for maintaining subtasks. It should also be noticed, in Figure 3(a), that the performance of *K-Adapt* declines more sharply than our *Learn-Adapt*, which implies that the scalability of *Learn-Adapt* is better than *K-Adapt*. Figure 3(b) shows the time consumption of both *Learn-Adapt* and *K-Adapt* in each simulation task allocation run, which is constituted by 15000 tasks. As expected, the running of *K-Adapt* costs less time than *Learn-Adapt*. This can be explained that *Learn-Adapt* requires a time consuming learning round, as

other learning algorithms doing, and agents using *Learn-Adapt* have to take more reward evaluation attributes into account, as described in Subsection 3.1, in order to choose an optimal action to change their relations. However, the gap of time consumption between *learn-Adapt* and *K-Adapt* is only in milliseconds level, which should be acceptable in many real cases. Figure 3(c) displays that with the average number of neighbours of each agent rising, the performance of both *Learn-Adapt* and *K-Adapt* improves. Since more neighbours can effectively reduce the length of delegation chain of each subtask, communication cost and management cost for maintaining subtasks of each agent could be lowered as well, which will lead to the improvement of entire network profit.

In summary, the performance of our method is around 80% $\sim$ 90% of the upper bound centralised allocation method, and on average 5% better than the *K-Adapt* method in small sized network (50 agents). In larger agent networks, although the performance of both methods reduces, *Learn-Adapt* still achieves relatively higher performance than *K-Adapt*. Thereby, it has been proven that our learning method can achieve higher performance for self-organisation than meta-reasoning to some extent, and it is believed in many scenarios that the improvement of performance and scalability deserves a little growth of time consumption.

5 Conclusion

This paper introduced a decentralised self-organisation mechanism which aims to adapt structural relations among agents in a network to achieve efficient task allocation. By using this mechanism, a pair of agents can independently evaluate each available action and jointly make a decision about taking an action to change their relation. Since this mechanism is decentralised and continuous over time, it meets the principles of self-organisation defined by Serugendo et al. [1]. We also empirically demonstrated that the performance of our mechanism approaches to the centralised method and consistently outperforms *K-Adapt*.

This research can be exploited for task allocation in many complex systems where resources are distributed and agents are highly autonomous, such as agent-based grid systems, service-oriented computing and wireless sensor networks (WSN). For example, our mechanism could be applied for optimising packet routing in WSN, since packets are somewhat similar as tokens in our model. Although each packet in WSN has a pre-specified destination which is different from a token, our mechanism could still work with a few minor modifications. Other potential application domains include automatic systems which are capable of self-management, e.g. e-commerce and P2P information retrieval.

In the future, we plan to improve and test our mechanism in a dynamic environment where agents can leave and join at any time step, and new types of resources might be obtained or lost by agents over time. In addition, another interesting stream of future work is that three or more agents might jointly make a decision with regard to changing relations among them.

References

[1] Serugendo, G.D.M., Gleizes, M.P., Karageorgos, A.: Self-organization in multi-agent systems. The Knowledge Engineering Review 20(2), 165–189 (2005)
[2] Mathieu, P., Routier, J.C., Secq, Y.: Principles for dynamic multi-agent organizations. In: Kuwabara, K., Lee, J. (eds.) PRIMA 2002. LNCS (LNAI), vol. 2413, pp. 109–122. Springer, Heidelberg (2002)
[3] Horling, B., Benyo, B., Lesser, V.: Using self-diagnosis to adapt organizational structures. In: AGENTS 2001, Montreal, Quebec, Canada, pp. 529–536 (May 2001)
[4] Hoogendoorn, M.: Adaptation of organizational models for multi-agent systems based on max flow networks. In: IJCAI 2007, Hyderabad, India, pp. 1321–1326 (January 2007)
[5] Kamboj, S., Decker, K.S.: Organizational self-design in semi-dynamic environments. In: AAMAS 2007, Honolulu, Hawai'i, USA, pp. 1228–1235 (May 2007)
[6] Gaston, M.E., des Jardins, M.: Agent-organized networks for dynamic team formation. In: AAMAS 2005, Utrecht, Netherlands, pp. 230–237 (July 2005)
[7] Glinton, R., Sycara, K., Scerri, P.: Agent organized networks redux. In: AAAI 2008, Chicago, Illinois, USA, pp. 83–88 (July 2008)
[8] Abdallah, S., Lesser, V.: Multiagent reinforcement learning and self-organization in a network of agents. In: AAMAS 2007, Honolulu, Hawai'i, USA, pp. 172–179 (May 2007)
[9] Kota, R., Gibbins, N., Jennings, N.R.: Decentralised structural adaptation in agent organisations. In: AAMAS 2008 Workshop on Organized Adaption in Multi-Agent Systems, Estoril, Portugal, pp. 1–16 (May 2008)
[10] Kota, R., Gibbins, N., Jennings, N.R.: Self-organising agent organisations. In: AAMAS 2009, Budapest, Hungary, pp. 797–804 (May 2009)
[11] Xu, Y., Scerri, P., Yu, B., Okamoto, S., Lewis, M., Sycara, K.: An integrated token-based algorithm for scalable coordination. In: AAMAS 2005, Utrecht, Netherlands, pp. 407–414 (July 2005)
[12] Gomes, E.R., Kowalczyk, R.: Dynamic analysis of multiagent q-learning with e-greedy exploration. In: ICML 2009, Montreal, Canada, pp. 369–376 (June 2009)
[13] Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature 393, 440–442 (1998)

Improving Trading Systems Using the RSI Financial Indicator and Neural Networks

Alejandro Rodríguez-González[1], Fernando Guldrís-Iglesias[1], Ricardo Colomo-Palacios[1], Juan Miguel Gomez-Berbis[1], Enrique Jimenez-Domingo[1], Giner Alor-Hernandez[2], Rubén Posada-Gomez[2], and Guillermo Cortes-Robles[2]

[1] Universidad Carlos III de Madrid, Av. Universidad 30, Leganés, 28918, Madrid, Spain
{alejandro.rodriguez,fernando.guldris,ricardo.colomo,
juanmiguel.gomez}@uc3m.es
[2] Division of Research and Postgraduate Studies,
Instituto Tecnológico de Orizaba, México
{galor,rposada,gcortes}@itorizaba.edu.mx

Abstract. Trading and Stock Behavioral Analysis Systems require efficient Artificial Intelligence techniques for analyzing Large Financial Datasets (LFD) and have become in the current economic landscape a significant challenge for multi-disciplinary research. Particularly, Trading-oriented Decision Support Systems based on the Chartist or Technical Analysis Relative Strength Indicator (RSI) have been published and used worldwide. However, its combination with Neural Networks as a branch of computational intelligence which can outperform previous results remain a relevant approach which has not deserved enough attention. In this paper, we present the Chartist Analysis Platform for Trading (CAST, in short) platform, a proof-of-concept architecture and implementation of a Trading Decision Support System based on the RSI and Feed-Forward Neural Networks (FFNN). CAST provides a set of relatively more accurate financial decisions yielded by the combination of Artificial Intelligence techniques to the RSI calculation and a more precise and improved upshot obtained from feed-forward algorithms application to stock value datasets.

Keywords: Neural Networks, RSI Financial Indicator.

1 Introduction

There has been growing interest in Trading Decision Support Systems in recent years. Forecasting the price movements in stock markets has been a major challenge for common investors, businesses, brokers and speculators. The stock market is considered as a high complex and dynamic system with noisy, non-stationary and chaotic data series [1], and hence, difficult to forecast [2]. However, despite its volatibility, it is not entirely random [3], instead, it is nonlinear and dynamic [4] or highly complicated and volatile [5]. Stock movement is affected by the mixture of two types of factors [6]: determinant (e.g. gradual strength change between buying side and selling side) and random (e.g. emergent affairs or daily operation variations).

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 27–37, 2010.

Wen et al. [1], argues that the study of the stock market behavior is a hot topic, because if it is successful, the result will yield fruitful rewards. Thus, it is obvious that predicting stock market movement is the long cherished desire of investors, speculators, and industries. However, this market is extremely hard to model with any reasonable accuracy [2]. Prediction of stock price variation is a very difficult task and the price movement behaves more like a random walk and time varying [7].

However, despite the referred complexity, many factors, including macroeconomic variables and stock market technical indicators, have been proven to have a certain level of forecast capability on stock market during a certain period of time. One of the tools for the financial practice is technical analysis, also known as "charting". According to Leigh et al. [8] Charles Dow developed the original Dow Theory for technical analysis in 1884 revisited by Edwards and Magee [9] more than a century before. Technical analysis studies the historical data surrounding price and volume movements of the stock by using charts as the primary tool to forecast future price movements. In recent years, and in spite of several critics [10], technical analysis has proven to be powerful for evaluating stock prices and is widely accepted among financial economists and brokerage firms.

The paper consists of five sections and is structured as follows. Section 2 surveys the relevant literature about technical analysis and its intersection with soft computing. Section 3 discusses the main features of CAST including conceptual model, algorithmic and architecture. Section 4 describes the evaluation of the tool performed. Finally, the paper ends with a discussion of research findings, limitations and concluding remarks.

2 Related Works

Stock price prediction using soft intelligence methods is not new. To solve the nonlinear problem and improve stock price evaluation, many researchers have focused on Technical Analysis by using advanced mathematics based techniques [11]. Along with the development of Artificial Intelligence, more researchers have tried to build automatic decision-making systems to predict stock market [12]. Soft computing techniques such as fuzzy logic, neural networks, and probabilistic reasoning draw most attention because of their abilities to handle uncertainty and noise in stock market.

White [13] was the first to use neural networks for market forecasting in the late 80s. In the early 90s Kimoto, Asakawa, Yoda, and Takeoka [14] used several learning algorithms and prediction methods for developing a prediction system for the Tokyo Stock Exchange Prices Index. Trippi and DeSieno [15] combined the outputs of individual networks using boolean operators to produce a set of composite rules. Other approaches can be found in various works from that decade using artificial neural networks [16] or computational intelligence [17].

As stated before, CAST (Chartist Analysis System for Trading) is based on the use of an improved version of RSI, one of the leading technical analysis indexes. RSI as a part of diverse calculations and formulas is commonly present in the soft computing research [18]. However, using soft computing methods in getting iRSI calculations is

a research task with no presence in the literature. Seeking this new research scenario in this work is proposed CAST, a system that uses generalized feedforward neural network to perform RSI improved calculations.

3 The CAST System

The idea of the system developed is to create a trading system based on technical or chartist analysis. Concretely, the main idea is to use one of the most used financial indicators, namely RSI. RSI is a financial technical analysis momentum oscillator measuring the velocity and magnitude of directional price movement by comparing upward and downward close-to-close movements. Momentum measures the rate of the rise or fall in stock price. Is the momentum increasing in the "up" direction, or is the momentum increasing in the "down" direction. Wilder [19] established that the more accurate value for N to calculate the best RSI is 14 because it was the half of the lunar cycle. However, depending on the market, the company and other factors, value 14 is not always the best value to calculate RSI. For this reason, in this paper is described a system capable of predict RSI values for a concrete market instead for a concrete company. The main idea is trying to predict the market behavior, concretely RSI behavior (as a collection of companies) and particularize for a concrete company using some correction factors. The system is divided in the following modules: neural network module, trading system module, RSI manager and generator module and heuristic module.

3.1 Neural Network Module

The Neural Network Module is the responsible of provide the RSI values that will be used to decide if an investor should invest in a certain company. The network used is a generalized feed-forward network [20]. The advantage of the generalized Feed Forward (FF) network is the ability to project activities forward by bypassing layers. The result is that the training of the layers closer to the input becomes much more efficient. Hereafter are showed the configuration of the network:

In first place, the input values of the network:

Table 1. Neural Network Input Values

Input Values	Explanation
IBEX35 Action Value	Value of the market for a concrete day.
RSI(9)	Calculus of RSI value using N = 9.
RSI(14)	Calculus of RSI value using N = 14.
RSI(30)	Calculus of RSI value using N = 30.
RSI Optimal	Calculus of optimal RSI using Heuristic

The topology of the network used was divided in an input layer, one hidden layer and an output layer. The next table shows the neurons set to each layer.

Table 2. Neural Network Configuration

Layer	Input Neurons	Output Neurons	Activation Function
Input	5	15	Laguarre (3 Taps, 1 Tap Delay)
Hidden	15	10	TanH
Output	10	1	Bias

The momentum value associated to the layers of the networks was set to 0.7. Following are showed the training values used in the network:

Table 3. Neural Network training set

Number of input data	Cross Validation	Training
4061 (> 16 Years of Financial Data)	20% of Input Data	10% of Input Data

The number of epochs set in order to train the network was specified to 10,000 epochs establishing that the network must stop learning after 500 epochs without improvement in cross-validation error.

3.2 Trading System Module

This module analyzes the result given by neural network module. When a particular query is formulated to the system, it takes the actual values of the market and builds a query to the neural network. It is important to stand out that in this case the predictions made by the system affect only to the IBEX35 (Spanish) market and not to a concrete company. However, in the heuristic section, some correction values will be shown in order to see that is possible to adapt the heuristic method to make predictions directly to the companies. The analysis made by the trading system is simple. It takes the value given by the neural network (RSI) and compares it with two extreme values.

- If RSI value is higher than 70 the decision that trading system will return is a sell signal. This value can be adapted and in some cases the value will be set to 65 instead of 70.
- If RSI value is lower than 30 the decision that trading system will return is a buy signal. This value can be adapted and in some cases the value will be set to 35 instead of 30.

3.3 RSI Manager and Generator Module

The RSI Manager and generator is the module in charge of manages and generates the RSI Values. This module calculates RSI(N) values where $N \in [5,35]$ and is used to train the network or to query it. However, there is another RSI estimation that must be done: Optimal RSI. There are two ways to calculate the Optimal RSI.

- **Optimal RSI for Market (IBEX35) prediction:** The calculus is provided by heuristic function (see next section).
- **Optimal RSI for Company (IBEX35) prediction:** In the context of company prediction Optimal RSI calculus is done taking into account the values that actions has in all the period without any modification, so, partial calculated RSIs are not stored and always is used the RSI of the day before and the actual day without taking into account previous RSIs.

3.4 Heuristic Module

The heuristic module is in charge of managing the different formulas that provide the heuristic used to generate the optimal values for RSI indicator. As was commented in the previous section exists more than one way to calculate the Optimal RSI value.

Optimal RSI for Market (IBEX35) prediction:

The formula of this heuristic is the next:

$$RSI = C1 + C2 * IBEX35 + C3$$

Where:

- IBEX35 represents the current value of the market on the actual date.
- C1 is a correction parameter set to -206.1103731082200000
- C2 is a correction parameter set to 0.0213265642264200
- C3 is a correction parameter set to 0.9947712774802800

The formula has been obtained using statistical techniques applied to the variables involved in the process. Concretely, a linear regression to relate the RSI values with IBEX35 stock market close values has been done. A total of 30 linear regressions were done to be able to collect possible values of RSI in function of the number of days. Each of these regressions used a different value of RSI when calculate itself with different time intervals. Concretely, it was permitted that the number of days used in the calculation could move between 5 and 35. The reason is to collect all the possible values used, because it is not normal that analysts use periods out of this range.

Each of the linear regressions calculated include a parameter called AR(1) to improve the specification of the model. Once results have been obtained, the models that did not fulfill the next requisites were discarded:

- Some of the independent variables is not statistical meaningful for explain the behavior of dependent variable.
- The probability of whole nullity of the model is zero.
- The goodness of fit is not enough good. In this sense, are considerate as no valid the models that cannot explain a 75% of the variation of the dependent variable.

- The model presents heteroscedasticity.
- The model presents autocorrelation.

Finally, in function of the mentioned criterions, has been discarded the models that include the RSI calculated with intervals of days equals to 5, 10, 12, 28, 30 and 35. With the rest of the models we proceed to calculate a unique equation that is its arithmetic mean, in order that the resulted heuristic is the more representative value of all the valid calculated models.

Optimal RSI for Company (IBEX35) prediction:

The formula of this heuristic is the next (explained in several steps):

$$rsiToday = C1 + (C2 * vMT) + C3$$
$$rsiYday = C1 + (C2 * vMY) + C3$$
$$restRSI = rsiToday - rsiYday$$
$$growthRate\ Company = (vCT - vCY) / vCY$$
$$growthRate\ Market = (vMT - vMY) / vMY$$
$$RSI = rsiYday + (restRSI * \frac{growthRateCompany}{growthRateMarket})$$

Where:

- vMT represents the value of the market today.
- vMY represents the value of the market yesterday.
- vCT represents the value of the company today.
- vCY represents the value of the company yesterday.
- C1, C2 and C3 are the same parameters with the same values.

4 Evaluation

The evaluation of the system consists in two parts. In the first place, the evaluation of the neural network and how they are able to predict will be evaluated. Particularly, in this evaluation the best configuration has been chosen for a he neural network set up to calculate better predictions using the iRSI. The second part consists of a definite query to the neural networks with a certain number of values and checks the signals that RSI value shots (buy or sell). The aim is to find out the accuracy of CAST.

4.1 Study 1: Evaluating Generalization of Neural Network

In the first part, several neural networks were evaluated with different configurations in order to choose the one that showed better results in terms of investment performance. The aim of this study is to find out which neural network configuration provides better results for CAST. Different neural networks configuration schemes were tested and their output was compared with real values in order to choose the one that reached better prediction rates. Next table shows the results from all neural networks tested:

Table 4. Results of neural networks

Name	# Input Parms	Parms	Test Ok	M-Test Ok (5 VOC)
RN1	32	AV, RSI[5-35]	7.881%	65.517%
RN2	2	AV	8.867%	75.862%
RN3	5	AV, RSI(9,14,30)	11.084%	80.296%
RN4	32	AV, RSI[5-35]	3.941%	37.931%
RN5	2	AV	7.882%	63.054%
RN6	5	AV, RSI(9,14,30)	7.389%	53.695%

Results show that RN3 configuration provides best results from the set analyzed. Using M-Test Column, results show that RN3 configuration can predict 80% (652 cases from 812) in an accurate way, which is a very good approach. With the objective of verifying if results presented statistical significant differences among neural networks configurations, the statistical method analysis of variance (ANOVA) was used to carry out analysis of variance among groups using the tool SPSS. The level of statistical significance was set at 0.05. It was used One-way ANOVA in order to test for differences among two or more independent groups (in our case six groups). The results of the test indicate that groups present significant differences indicated by the statistical value ($F(811)= 89.704$, $p<.05$). This circumstance implies that, from a statistical point of view, there is a difference among predictions. However, it is important to work out if there's a difference between best prediction configuration and real values. It was done using the statistical method Student's t-test (comparison of two means). The results of this test showed significant differences between real values and best predictions corresponding to RN3 ($t(811)=20.716$, $p<.05$). This circumstance reveals that there are ways to improve our solution either by better training or new setup definition, although it presents undeniable good results. In order to proceed with Study 2 the neural network that will be used as reference for further studies will be the one named RNAR3 for its remarkable capability of generalization according to its results.

4.2 Study 2: Evaluating Behavior of Neural Network in a Real Case

In Study 2, once chosen in Study 1 the best neural network configuration (RN3), it was tested applying a real market scenario in a given period of time. The aim of this study is to know if iRSI proposed in this paper gives better results than conventional RSI using 14 days as was suggested by Wilder. In Study 2 this method will be noted by RSI14. Given that CAST can be applied to two different scenarios, in the one hand whole market prediction (IBEX 35, in this case), and on the other hand, single company prediction (pertaining IBEX 35), there will be two main tests. The aim of the first test is to compare iRSI versus RSI14 for IBEX 35 stock market. The aim of the second test is to prove that iRSI can predict single company values pertaining IBEX 35 in a more accurate manner compared with RSI14. The comparison method, which is common to test 1 and test 2, is as follows: given a concrete day and IBEX 35 index (test 1) or Company (test 2), the index is performed (RSI14 and iRSI). It advises either "Sell" or "Buy". Once the action is performed the value related to this operation (of selling or buying) is compared with the final value after a period of time (from one

day to seven days). If the value is coherent with the prediction (lower if a sell command was send or higher if it was delivered a buy command) then this action increments a success counter for the method proved, RSI14 in this case. Final score will be (success operations) / (total operations).

The sample used in this test consisted in the data obtained in the period between the 16th December 2005 and the 27th October 2009, a total of 812 values. For Test 1, IBEX 35 values were used. For Test 2, fifteen companies of Spanish stock market (IBEX35) were used using 812 values per company, a total of 12,180 values.

TEST 1

Next table shows the results of the chosen neural network for iRSI prediction compared with RSI14 predicting IBEX 35 behavior:

Table 5. RSI14 vs iRSI

Period	RSI14	iRSI
1 Day	45.65%	**53.83%**
2 Days	43.00%	**55.46%**
3 Days	43.48%	**56.15%**
4 Days	42.27%	**58.47%**
5 Days	41.79%	**58.61%**
6 Days	41.79%	**58.61%**
7 Days	42.27%	**58.20%**

The results obtained show a better performance of iRSI compared to RSI14. All measures present higher values in iRSI and all of them are above 50%. On the other hand, none of the measures by RSI14 present measures above 46%. Higher success values for iRSI are present in 5 and 6 days period (58.61%). The first test performed in order to find out if there are significant differences between RSI14 and iRSI was a Student's t-test (comparison of two means). The test was performed for the predictions done without taking into account periods. The results of this test showed significant differences between iRSI and RSI14 ($t(11367)=15.407$, $p<.05$). This circumstance that can be easily inferred from results reveals that iRSI is a better guidance for investors in order to predict IBEX35 stock market. Applying this same test to pairs of predictions, results show in every case a better prediction as well as a significant difference between every RSI14 and iRSI pair. The second issue is to find out, on the one hand, if there are significant differences among periods as a whole in iRSI. To do so, it was used One-way ANOVA in order to test for differences among two or more independent groups (in this case seven groups). The result of this test pointed out that there are no significant differences among periods for iRSI ($F(5683)= 1.236$, $p>.05$). This asseveration can be inferred also from results, showing just slight differences among scores.

The application of neural networks to Ibex35 stock prediction is not new [21]. In these cases, neural networks provide a reasonable description of asset price movements, in our case, in which the aim is to improve RSI, results show an unquestionable empowerment of results compared with RSI14.

TEST 2

Finally, we show the results of the prediction using the neural network (RNAR3) adapted with the heuristic of RSI calculus using some companies of the IBEX35 market. Table 6 shows results of the application of RSI14 for selected enterprises pertaining IBEX 35 stock market:

Table 6. Results of RSI14 to selected enterprises

Company	RSI14 PR 1D	RSI14 PR 2D	RSI14 PR 3D	RSI14 PR 4D	RSI14 PR 5D	RSI14 PR 6D	RSI14 PR 7D
Enagas	52.36%	56.22%	54.51%	56.22%	57.51%	57.94%	55.36%
ACS	43.60%	46.71%	44.29%	41.18%	41.52%	41.18%	42.21%
Inditex	49.78%	48.89%	49.33%	49.78%	48.00%	45.33%	43.11%
Telecinco	44.30%	44.73%	45.15%	45.15%	40.93%	41.77%	41.77%
Santander	49.57%	50.00%	53.02%	55.60%	53.02%	50.43%	50.43%
Indra	58.38%	59.46%	62.70%	63.78%	65.95%	65.41%	65.41%
Abengoa	52.40%	50.92%	48.71%	50.18%	47.97%	49.45%	51.29%
Iberia	48.28%	49.43%	42.53%	43.30%	38.70%	41.38%	38.31%
Iberdrola	49.29%	46.79%	46.79%	49.29%	47.14%	44.64%	45.71%
Repsol	48.24%	51.37%	52.16%	50.98%	50.20%	53.33%	52.55%
Sacyr	48.37%	46.88%	49.85%	48.66%	45.10%	44.81%	43.03%
BBVA	47.58%	50.40%	50.81%	52.02%	49.60%	51.21%	50.00%
Banesto	42.97%	46.39%	45.25%	42.21%	41.06%	38.02%	39.92%
Telefónica	49.09%	48.18%	47.73%	45.91%	46.82%	46.82%	46.82%
Abertis	55.90%	52.31%	52.31%	49.74%	50.26%	49.74%	47.69%

Next table shows results of the application of iRSI using RNAR3. Figures in bold mean better results for iRSI than RSI14.

Table 7. Results of iRSI using RNA3

Company	RSI14 PR 1D	RSI14 PR 2D	RSI14 PR 3D	RSI14 PR 4D	RSI14 PR 5D	RSI14 PR 6D	RSI14 PR 7D
Enagas	46.49%	48.43%	49.93%	48.43%	48.58%	48.43%	47.83%
ACS	**47.31%**	**50.75%**	**49.85%**	**50.90%**	**51.79%**	**51.49%**	**51.19%**
Inditex	**50.30%**	**52.99%**	**54.34%**	**54.94%**	**53.29%**	**52.10%**	**54.57%**
Telecinco	**49.41%**	**54.45%**	**55.49%**	**56.68%**	**56.97%**	**56.82%**	**56.23%**
Santander	**51.49%**	**52.99%**	**54.48%**	55.52%	**55.37%**	**55.52%**	**55.97%**
Indra	48.49%	53.02%	54.38%	55.89%	56.19%	55.44%	55.29%
Abengoa	**53.93%**	**53.78%**	**52.15%**	**52.59%**	**53.33%**	**52.15%**	**52.74%**
Iberia	**49.93%**	**49.93%**	**48.14%**	**50.07%**	**49.48%**	**51.12%**	**51.56%**
Iberdrola	**53.06%**	**52.61%**	**53.50%**	**53.50%**	**54.25%**	**52.61%**	**53.35%**
Repsol	**56.54%**	**53.83%**	50.38%	**52.33%**	**52.78%**	51.43%	52.48%
Sacyr	**55.44%**	**55.14%**	**55.14%**	**56.80%**	**57.40%**	**57.85%**	**59.67%**
BBVA	**54.40%**	**52.91%**	**54.99%**	**54.99%**	**54.55%**	**54.55%**	**54.40%**
Banesto	**52.02%**	**53.36%**	**53.66%**	**54.41%**	**55.46%**	**54.26%**	**55.31%**
Telefónica	48.80%	**49.10%**	**49.40%**	**48.05%**	**47.46%**	**47.46%**	**47.90%**
Abertis	52.82%	51.48%	51.63%	**53.41%**	**53.56%**	**54.45%**	**54.38%**

As it can be observed in the result section of the evaluation the use of the neural networks using heuristic formulas calculated by linear regression of financial factors for training can improve RSI14.

5 Conclusions and Future Work

The current work describes a research work about the generation of RSI values to create systems capable of generate automated or semi-automated investments on certain companies in the IBEX35 Spanish stock market. In this paper the main work is based in the study case of generate a heuristic for a concrete market, and apply some corrections factor in order to be able to generate good investment results for a concrete companies of the sector. This work was only based in RSI financial indicator and the heuristic methods applied where generated to create a single heuristic formula for IBEX35 stock market. The current work proposes four types of initiatives which should be explored in future research. In the first place, our future work plans to generate heuristic for each company analyzing their data. In the second place, extend the application of iRSI to a broader sample: more companies pertaining IBEX 35, more indexes both national and international and, of course, as stated before, more time frame. In the third place, tune iRSI to adapt it to a momentum in the market. Finally, expand the research to investigate a broader technical analysis indexes like MACD (Moving Average Convergence / Divergence) financial indicator or Bollinger Bands.

Acknowledgments. This work is supported by the Spanish Ministry of Industry, Tourism, and Commerce under the EUREKA project SITIO (TSI-020400-2009-148), SONAR2 (TSI-020100-2008-665 and GO2 (TSI-020400-2009-127). Furthermore, this work is supported by the General Council of Superior Technological Education of Mexico (DGEST). Additionally, this work is sponsored by the National Council of Science and Technology (CONACYT) and the Public Education Secretary (SEP) through PROMEP.

References

1. Wen, Q., Yang, Z., Song, Y., Jia, P.: Automatic stock decision support system based on box theory and SVM algorithm. Expert Systems with Applications 37(2), 1015–1022 (2010)
2. Wang, Y.F.: Mining stock prices using fuzzy rough set system. Expert System with Applications 24(1), 13–23 (2003)
3. Chiu, D.Y., Chen, P.J.: Dynamically exploring internal mechanism of stock market by fuzzy-based support vector machines with high dimension input space and genetic algorithm. Expert Systems with Applications 36(4), 1240–1248 (2009)
4. Hiemstra, C., Jones, D.: Testing for Linear and Nonlinear Granger Causality in the Stock Price-volume Relation. Journal of Finance 49(5), 1639–1664 (1994)
5. Black, A.J., Mcmillan, D.G.: Non-linear predictability of value and growth stocks and economic activity. Journal of Business Finance and Accounting 31(3/4), 439–474 (2004)
6. Bao, D., Yang, Z.: Intelligent stock trading system by turning point confirming and probabilistic reasoning. Expert Systems with Applications 34(1), 620–627 (2008)

7. Chang, P.C., Liu, C.H.: A TSK type fuzzy rule based system for stock price prediction. Expert Systems with Applications 34(1), 135–144 (2008)
8. Leigh, W., Modani, N., Purvis, R., Roberts, T.: Stock market trading rule discovery using technical charting heuristics. Expert Systems with Applications 23(2), 155–159 (2002)
9. Edwards, R., Magee, J.: Technical analysis of stock trends, 7th edn. Amacom, New York (1997)
10. Malkiel, B.G.: A random walk down wall street. Norton & Co., New York (1995)
11. Wang, J.L., Chan, S.H.: Stock market trading rule discovery using two-layer bias decision tree. Expert Systems with Applications 30(4), 605–611 (2006)
12. Kovalerchuk, B., Vityaev, E.: Data mining in finance: advances in relational and hybrid methods. Kluwer Academic, Dordrecht (2000)
13. White, H.: Economic prediction using neural networks: The case of IBM daily stock returns. In: Proceedings of the 2nd Annual IEEE Conference on Neural Networks, II, pp. 451–458 (1988)
14. Kimoto, T., Asakawa, K., Yoda, M., Takeoka, M.: Stock market prediction system with modular neural network. In: Proceedings of the International Joint Conference on Neural Networks, San Diego, CA, pp. 1–6 (1990)
15. Trippi, R.R., DeSieno, D.: Trading equity index futures with a neural network. Journal of Portfolio Management 19(1), 27–33 (1992)
16. Aiken, M., Bsat, M.: Forecasting market trends with neural networks. Information Systems Management 6(4), 42–48 (1994)
17. Wang, L.P., Fu, X.J.: Data Mining with Computational Intelligence. Springer, Heidelberg (2005)
18. Yao, J., Herbert, J.P.: Financial time-series analysis with rough sets. Applied Soft Computing 9(3), 1000–1007 (2009)
19. Wilder Jr., J.W.: New Concepts in Technical Trading Systems. Hunter Publishing Company, Greensboro (1978)
20. Arulampalam, G., Bouzerdoum, A.: A generalized feedforward neural network architecture for classification and regression. Neural Networks 16(5-6), 561–568 (2003)
21. Fernández-Rodríguez, F., González-Martel, C., Sosvilla-Rivero, S.: On the profitability of technical trading rules based on artificial neural networks: Evidence from the Madrid stock market. Economics Letters 69(1), 89–94 (2000)

Balanced Student Partitioning to Promote Effective Learning: Applications in an International School

Wenbin Zhu[1,3], Hu Qin[2,⋆], Andrew Lim[2], and Zhou Xu[3]

[1] Department of Computer Science and Engineering,
The Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, Hong Kong
i@zhuwb.com
[2] Department of Management Sciences,
City University of Hong Kong,
Tat Chee Ave, Kowloon Tong, Hong Kong
{tigerqin,lim.andrew}@cityu.edu.hk
[3] Department of Logistics and Maritime Studies,
The Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong
lgtzx@polyu.edu.hk

Abstract. In this paper, we describe a system that our team developed to solve the problem of heterogeneously partitioning students into classes for the Singapore International School based in Hong Kong. This problem has multiple objectives such as to achieve similar class sizes, similar gender ratios among all classes, each student having at least one old classmate of the same gender, conflict avoidance among students, and similarity of score distribution curves. We proved that this problem is extremely hard and provided an example to show that the number of feasible solutions is astronomical for only medium size cases. We devised and implemented a simulated annealing (SA) algorithm to solve this problem. Our experimental results based on real application data indicate that our SA algorithm is able to improve the quality of the school's partitioning solutions and clearly meets all objectives set out by the client.

Keywords: student partitioning; simulated annealing; multiple objective.

1 Introduction

The problem investigated in this paper is derived from a project that our team was awarded by the Singapore International School Hong Kong (SISHK). It is an international school comprising an elementary and secondary section, and is run by the government of Singapore, following the Singapore curriculum and

⋆ Corresponding author.

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 38–48, 2010.

education philosophy. Because SISHK only recently established the secondary section, the school currently has eight grades and over 700 students. Every academic year (starting from September) SISHK systematically partitions students based on their academic performance records of the previous year. The problem is addressing how to partition students into classes with the aim of optimizing multiple objectives which are detailed in the later sections of this paper.

Presently, there are two main policies when partitioning students [1]: (1) homogeneous partitioning (also called ability grouping or tracking) which is the practice of placing students into different classes based on prior academic results; (2) heterogeneous partitioning (also called non-tracking) is simply that of mixed-abilities partitioning.

The rationale for having homogeneously partitioned students is that the school is better able to offer courses tailored to different abilities, i.e., schools may have special remedial classes for low achievers and enrichment courses for gifted and talented students and the teachers can teach and manage the classes more easily. Moreover, lower-achieving students may feel more comfortable and participate more when they are grouped with peers of similar ability and higher-achieving students can maintain interests without being impeded by slower learners.

Although homogeneous partitioning seems fair, non-coincidentally, racial and ethnic minorities, or those with lower socioeconomic statuses usually end up in the lower education tracks, thereby being placed in the lower-achieving classes and inevitably receiving a lower level of instruction. However, the number of average students and low achievers is much larger than that of high achievers and because homogeneous partitioning is especially beneficial to the most talented students, this deprives most of the students of the opportunity to strive for greater success. On another note, students in low-achieving classes are often taught by inexperienced or incapable teachers (although it is unknown why this is the case). Furthermore, the absence of smarter students may have a detrimental effect since their "smarter" peers are not there to encourage and stimulate them. Most likely, negative labeling shatters confidence levels, thereby reinforcing self-fulfilling prophecies that these students are slow learners. Homogeneous partitioning thus widens the achievement gap between students in high- and low-achieving classes and it does not offer equal educational opportunities to all.

Diversity of views, cultures and experiences provides an invaluable element in educational institutes and heterogeneous partitioning is viewed as one form of diversity, allowing beneficial and meaningful interactions with students of differing abilities. These interactions provide students with an appreciation that there is much more out there that they have yet to discover (of ability, of interest, of racial and ethnic backgrounds, etc). Heterogeneous partitioning is generally viewed as more realistic and is a more favorable preparation for real life.

Since 1992, schools across the United States began switching from practices with unequal access to knowledge to those where all students are now provided with equal educational opportunities [1]. Similarly, the management at SISHK also favors the concept of heterogeneous partitioning. However, although SISHK considers heterogeneous partitioning as the better of the two options, there still

exists a problem in partitioning its students. Manually partitioning students is simple when the numbers of students and partitioning criteria are small. However, as with the case of SISHK, it is a relatively large-size school with a rather varied combination of factors to consider such as academic abilities, race, ethnicity, gender to name a few. The more the variations the more significant the complexities get. Given the number of students and classes being n and p respectively and assuming the student number in all classes are equal, i.e., n/p is an integer, according to the theory of permutation we can derive that the number of possible combinations is

$$\frac{n!}{p!((n/p)!)^p}$$

Some examples are shown in Table 1.

Table 1. Number of Possible Combinations

Number of students	Number of classes	Possible combinations
4	2	3
10	2	126
15	3	126,126
20	2	92,378
30	3	925,166,131,890
60	3	96,305,202,413,079,303,971,977,650

We can easily find that with just 60 students and 3 different classes, there are over ninety-six septillion (10^{24}) possible combinations. Consequently, for a school with similar size as SISHK, manual attempts at partitioning students are time-consuming and inaccurate and thus maintaining diversity is most likely impossible. To tackle this problem, our team developed an application we coined the Student Partitioning System (SPS) to help SISHK streamline the student partitioning process such that it can generate classes with diversity and similarities in its overall composition.

2 Problem Description and Formulation

There is a set of students $I = \{1, 2, \ldots, n\}$, a set of subjects $J = \{1, 2, \ldots, m\}$ and a set of classes $K = \{1, 2, \ldots, p\}$. Each student $i \in I$ has a score for each subject $j \in J$, denoted by s_{ij}. The gender of student $i \in I$ is represented by $g(i)$, where 1 denotes female and 0 denotes male. Most students have some old classmates or friends and occasionally some students have several "enemies" whom they often come into conflict with. For each student, we have two lists which contain names of old classmates of the same gender and enemies. All students are placed into p classes whilst fulfilling as optimally as possible the following requirements: (1) all classes have similar sizes; (2) gender ratios among all classes are close; (3) every

student in a class will have at least one old classmate of the same gender; (4) two students with conflict should not be assigned to the same class; (5) students' performance of each subject should be as similar as possible, i.e., for all classes, the average scores of each subject are similar, and the score distribution curves of each subject have a similar shape. The last requirement is of great importance for SISHK. This is because SISHK evaluates the performance of teachers based on the improvements of the students in the classes they teach, so the classes with similar score compositions are the basis of fair comparison.

To well expose the rest of this section, we introduce some decision variables. The binary decision variable x_{ik} equals 1 if student $i \in I$ is assigned to class $k \in K$ and equals 0 otherwise. Since one student must be assigned to only one class, we have $\sum_{k \in K} x_{ik} = 1$. The binary variable y_{ik} equals 1 if student $i \in I$ is assigned to class $k \in K$ and does not have old classmates of the same gender, and equals 0 otherwise. Obviously, $\sum_{k \in K} y_{ik} \leq 1$. Integral decision variable z_k is the number of student pairs having conflict in class k.

Similar class sizes can be realized by:

$$\min \left(\max_{k \in K} \left\{ \sum_{i \in I} x_{ik} \right\} - \min_{k \in K} \left\{ \sum_{i \in I} x_{ik} \right\} \right) \tag{1}$$

If all class sizes satisfy inequality (2), no improvement can be gained.

$$\lfloor n/p \rfloor \leq \sum_{i \in I} x_{ik} \leq \lceil n/p \rceil \tag{2}$$

Even female distribution can be obtained by:

$$\min \left(\max_{k \in K} \left\{ \sum_{i \in I} x_{ik} g(i) \right\} - \min_{k \in K} \left\{ \sum_{i \in I} x_{ik} g(i) \right\} \right) \tag{3}$$

Analogously, if the number of female students in each class satisfies the inequality (4), no improvement can be gained.

$$\lfloor \sum_{i \in I} g(i)/p \rfloor \leq \sum_{i \in I} x_{ik} g(i) \leq \lceil \sum_{i \in I} g(i)/p \rceil \tag{4}$$

Obviously, under the condition of similar class size and even female distribution, the number of male students in each class follows the same principle. We try to ensure each student has at least one old classmate of the same gender in his or her class and avoid student pairs with conflicts by expression (5) and (6).

$$\min \max_{k \in K} \left\{ \sum_{i \in I} y_{ik} \right\} \tag{5}$$

$$\min \max_{k \in K} \{ z_k \} \tag{6}$$

For subject j, we can compute the average score of each class k. The difference between the maximum average and minimum average among all classes can be

used as an indicator to measure the dissimilarities of that subject. To achieve similar average scores in each subject for all classes, we could try to minimize:

$$\max_{j\in J}\left\{w_j^5\Big(\max_{k\in K}\big\{\frac{\sum_{i\in I}x_{ik}s_{ij}}{\sum_{i\in I}x_{ik}}\big\}-\min_{k\in K}\big\{\frac{\sum_{i\in I}x_{ik}s_{ij}}{\sum_{i\in I}x_{ik}}\big\}\Big)\right\} \tag{7}$$

where w_j^5 is the penalty cost associated with subject j.

For subject j, we sort students in descending order according to their scores and use $d_j(l)$ to represent the l-th highest score. As a result, we get a score list and after separating this list into $\lceil n/p\rceil$ segments, each of which excluding the last one contains exactly p elements, we could compute an average score $a_j(h)$ for each list segment h, where $1\le h\le\lceil n/p\rceil$. With these average scores, we can construct a model class (dummy class) with $\lceil n/p\rceil$ students, each of which has score $a_j(h)$. Then, we sort students in class k in descending order according to their scores of subject j and use $d_{jk}(l)$ to represent the l-th highest score. We can define the difference between class k and the model class for subject j as follows

$$\sqrt{\sum_{l=1}^{q_i}(a_j(l)-d_{jk}(l))^2} \tag{8}$$

where $q_i=\min\{\sum_{i\in I}x_{ik},\lceil n/p\rceil\}$. To achieve similar score distribution curves for each subject for all classes, we attempted to minimize:

$$\max_{j\in J}\left\{w_j^6\max_{k\in K}\left\{\sqrt{\sum_{l=1}^{q_i}(a_j(l)-d_{jk}(l))^2}\right\}\right\} \tag{9}$$

where w_j^6 is the penalty cost associated with subject j.

With the aid of penalty costs, we simultaneously consider multiple objectives by following the cost function below:

$$\begin{aligned}
&w^1\Big(\max_{k\in K}\Big\{\sum_{i\in I}x_{ik}\Big\}-\min_{k\in K}\Big\{\sum_{i\in I}x_{ik}\Big\}\Big)\\
&+w^2\Big(\max_{k\in K}\Big\{\sum_{i\in I}x_{ik}g(i)\Big\}-\min_{k\in K}\Big\{\sum_{i\in I}x_{ik}g(i)\Big\}\Big)\\
&+w^3\max_{k\in K}\Big\{\sum_{i\in I}y_{ik}\Big\}+w^4\max_{k\in K}\{z_k\}\\
&+\max_{j\in J}\left\{w_j^5\Big(\max_{k\in K}\big\{\frac{\sum_{i\in I}x_{ik}s_{ij}}{\sum_{i\in I}x_{ik}}\big\}-\min_{k\in K}\big\{\frac{\sum_{i\in I}x_{ik}s_{ij}}{\sum_{i\in I}x_{ik}}\big\}\Big)\right\}\\
&+\max_{j\in J}\left\{w_j^6\max_{k\in K}\left\{\sqrt{\sum_{l=1}^{q_i}(a_j(l)-d_{jk}(l))^2}\right\}\right\}
\end{aligned} \tag{10}$$

where w^1,w^2,w^3,w^4 are penalty costs in association with expression (1), (3), (5) and (6), respectively.

2.1 Computational Complexity

Theorem 1. *There is no pseudo-polynomial algorithm that can achieve a constant approximation ratio to minimize (7) unless $P = NP$.*

Proof. By contradiction, suppose $P \neq NP$ and there is an Algorithm A that can achieve a constant approximation ratio of ρ to minimize (7) in pseudo-polynomial time. We are going to show that A can solve the following well-known strongly NP-hard problem in polynomial time.

3-Partition. given a set $S = \{s_1, s_2, ..., s_n\}$ of $n = 3p$ positive integers, can S be partitioned into p subsets S_1, S_2, S_p such that the sum of the numbers in each subset is equal? [2]

Given any instance of 3-Partition, consider the following instance of SISHK, with n students who have attended one subject only. Each student i for $1 \leq i \leq n$ has a score s_i for this subject. It is easy to see that the instance of 3-Partition has a feasible partition if and only if the instance of SISHK has a partition with (7) equal zero, if and only if A returns a solution to the instance of SISHK with (7) equal zero. Thus A can be applied to solve the 3-Partition in pseudo-polynomial time, implying $P = NP$. This leads to a contradiction.

Theorem 2. *Even if p is a constant, there is no polynomial algorithm that can achieve a constant approximation ratio to minimize (7) unless $P = NP$.*

Proof. This can be proved by a similar reduction from the following well-known NP-hard problem.

Partition. given a set $S = \{s_1, s_2, ..., s_n\}$ of $n = 2k$ positive integers, can S be partitioned into 2 subsets S_1 and S_2 such that the sum of the numbers in each subset is equal?

3 Solution Procedure – Simulated Annealing Algorithm

To solve the problem, we first used a simple greedy algorithm shown in Algorithm 1 to generate an initial solution S_0 and then applied the simulated annealing (SA) algorithm shown in Algorithm 2 to find improved solution S. SA is a well-known metaheuristics armed with randomized neighborhood search for the global optimization problem [3]. It can quickly achieve a good approximate solution to the global optimum of a given objective function in a large search space. From Table 1, we know that the number of feasible solutions is astronomically large when the problem size increases to a medium level, so we chose SA for this project. From the computational results which are available in the later section, we can draw the conclusion that SA can solve student partitioning problem well and generate solutions satisfying all needs of our client.

Our SA uses a simple cooling scheme. The outer loop (line 6) controls the cooling process. In each iteration, the temperature is multiplied by cooling ratio r, a number very close to but smaller than one. The inner loop (line 8) tries to

Algorithm 1. Greedy Algorithm for Initial Solution (n students, p classes)

1: Sort male students in descending order according to their individual average scores of all subjects;
2: Assign 1st student to class 1, 2nd to class 2, ..., pth to class p, $(p+1)$th to class p, $(p+2)$th to class $p-1$, ..., $2p$th to class 1, ...;
3: Sort female students in descending order according to their individual average scores of all subjects;
4: Continue with the last class in step 2 and assign students in similar manner as assigning male students;

Algorithm 2. The Simulated Annealing Algorithm

1: Generate an initial solution S_0 by running greedy algorithm ;
2: T_0, T_e, $r(0 < r < 1)$ are initial temperature, termination temperature and cooling ratio;
3: The maximum number of successful operations is L and the maximum number of attempts is N, at given temperature;
4: S_b is the currently best known solution and S is the current solution;
5: $t \leftarrow T_0$, $S_b \leftarrow S_0$;
6: **while** $t > T_e$ **do**
7: $i \leftarrow 0,\ n \leftarrow 0$;
8: **while** $i < L$ *and* $n < N$ **do**
9: Randomly pick two students from different classes and swap;
10: $\Delta \leftarrow$ objective value after swap minus objective value before swap;
11: **if** $\Delta < 0$ **then**
12: Accept the swap and update S; update S_b if $S < S_b$;
13: **else**
14: Accept the swap with a probability $e^{-\Delta/t}$; otherwise undo the swap;
15: **end if**
16: **if** S is changed **then**
17: $i \leftarrow i + 1$, $n \leftarrow 0$;
18: **end if**
19: $n \leftarrow n + 1$
20: **end while**
21: $t \leftarrow t \cdot r$
22: **end while**

find the equilibrium. We simply consider L successful swaps or N consecutive unsuccessful attempts as reaching equilibrium.

Since its founding, SISHK has relied on experienced staffs to place students. After analyzing the manual solutions of previous years, the principal believed that there should be space for improving the quality of the partitioning results. In March 2008, the principle initiated this project with our team to review their current manual partitioning process and study how to achieve better solutions.

At the stage of data collection, the difficulty laid in clarifying student pairs with conflict because some conflicts were undiscovered. The teachers spent nearly a month communicating with students continuously in order to depict more

accurately the relationship map among students. In the manual partitioning process, they did not deliberately take avoiding conflicts into account. Fortunately, other data required in the problem such as the scores of each student in each subject were easily retrieved. It took one month for us to implement the first version of the system. In the following month, we began testing our system based on real data and tuned the parameters accordingly. After comparing the solution output by our system with the results generated manually based on data of year 2007, SISHK management was convinced by our results and decided to accept our SPS to replace its manual process. At the end of August 2008, SISHK ran our SPS system to partition all students into classes using data from the previous academic year. Some of the results are presented in the following section.

4 Computational Results

To measure the performance of our SA and convince the principal to accept our system, we asked SISHK to provide us with real data and relative results generated manually. The information based on which we ran our system includes: (1) SISHK has 6 primary grades and 2 secondary grades, which are represented by $P1, P2, \ldots, P6$ and $S1, S2$. Since grade $P1$ does not have prior records, we could not apply our system to place students in this grade. (2) the number of classes in grade $P2, \ldots, P6, S1, S2$ are 5, 6, 5, 5, 3, 3, 2. (3) the number of students in grade $P2, \ldots, P6, S1, S2$ are 108, 142, 118, 119, 89, 78, 41. (4) there are four core subjects, namely are English, Chinese, Mathematics and Science. (5) each student has prior scores of his or her subjects. Students in $P1, P2, P3$ do not have records of Science. (6) for each student, we know all his or her old classmates of the same gender and "enemies".

We implemented the system in Java and ran it on an Intel Xeon(R) 2.66 GHz server with 3GB RAM. Our system includes a number of controlling parameters.

	Manual				SA			
	Female(F)	Male(M)	M-F	M+F	Female(F)	Male(M)	M-F	M+F
P2/C1	10	12	2	22	9	12	3	21
P2/C2	7	13	6	20	9	12	3	21
P2/C3	8	12	4	20	10	12	2	22
P2/C4	10	12	2	22	10	12	2	22
P2/C5	12	12	0	22	9	13	4	21
Standard deviation	1.95	0.45	2.28	1.10	0.55	0.45	0.84	0.55
Max-Min	5	1	6	2	1	1	2	1

Fig. 1. Distribution of male and female students for $P2$

	Manual			SA		
	F	M	M-F	F	M	M-F
P2	5	1	6	1	1	2
P3	4	1	4	1	1	2
P4	6	5	11	1	1	2
P5	1	2	3	1	1	2
P6	4	5	9	1	0	1
S1	3	3	6	1	1	2
S2	1	2	3	1	0	1

Fig. 2. Gender *Max-Min* for all grades

	Manual				SA			
	English	Chinese	Mathematics	Science	English	Chinese	Mathematics	Science
P2/C1	83.8	89.5	81.0	N/A	84.2	90.1	82.0	N/A
P2/C2	82.6	90.7	80.7	N/A	84.1	90.1	81.9	N/A
P2/C3	83.4	89.7	81.0	N/A	84.2	90.2	81.9	N/A
P2/C4	85.9	90.1	84.3	N/A	84.2	90.2	82.0	N/A
P2/C5	85.0	90.7	82.3	N/A	84.2	90.1	81.9	N/A
Standard deviation	1.2	0.5	1.3	N/A	0.0	0.0	0.0	N/A
Max - Min	3.2	1.2	3.6	N/A	0.1	0.1	0.1	N/A

Fig. 3. Average scores for $P2$

	Manual				SA			
	English	Chinese	Mathematics	Science	English	Chinese	Mathematics	Science
P2	3.2	1.2	3.6	N/A	1.9	2.2	3.3	N/A
P3	4.2	2.3	7.2	N/A	2.4	1.7	3.0	N/A
P4	6.8	6.8	8.1	4.1	2.3	2.0	2.6	2.2
P5	5.1	3.6	4.8	2.8	0.9	2.1	2.7	2.5
P6	0.8	4.3	0.1	4.4	0.3	1.7	2.1	0.7
S1	1.0	1.1	2.6	1.3	1.3	2.1	1.2	1.7
S2	1.1	0.3	1.5	0.4	0.9	3.2	0.3	0.3

Fig. 4. Score *Max-Min* for all grades

After conducting some preliminary experiments, we set the parameters to the following values: $T_0 = 2$, $T_e = 00001$, $r = 0.999$, $L = 50$ and $N = 500$; $w_1 = 100$, $w_2 = 100$, $w_3 = 1000$ and $w_4 = 1000$; for each $j \in J$, $w_5^j = 50$ and $w_6^j = 10$.

Compared with the manual partitioning results, solutions generated by SA were much better for all seven grades. Using $P2$ as an example, experimental

results are shown in Fig. 1, where *Max* and *Min* mean the maximum and minimum number of students, male students, or female students among all classes, and $P2/Ck$ denotes class k in $P2$. In this figure, we find that students with both genders are distributed unevenly in the manual solution since *Max-Min* of F and $M - F$ are 5 and 6 while corresponding numbers generated by SA are 1 and 2. The smaller standard deviation also reveals that all classes have similar number of students for both genders in our SA solution. Similarly, we find that the

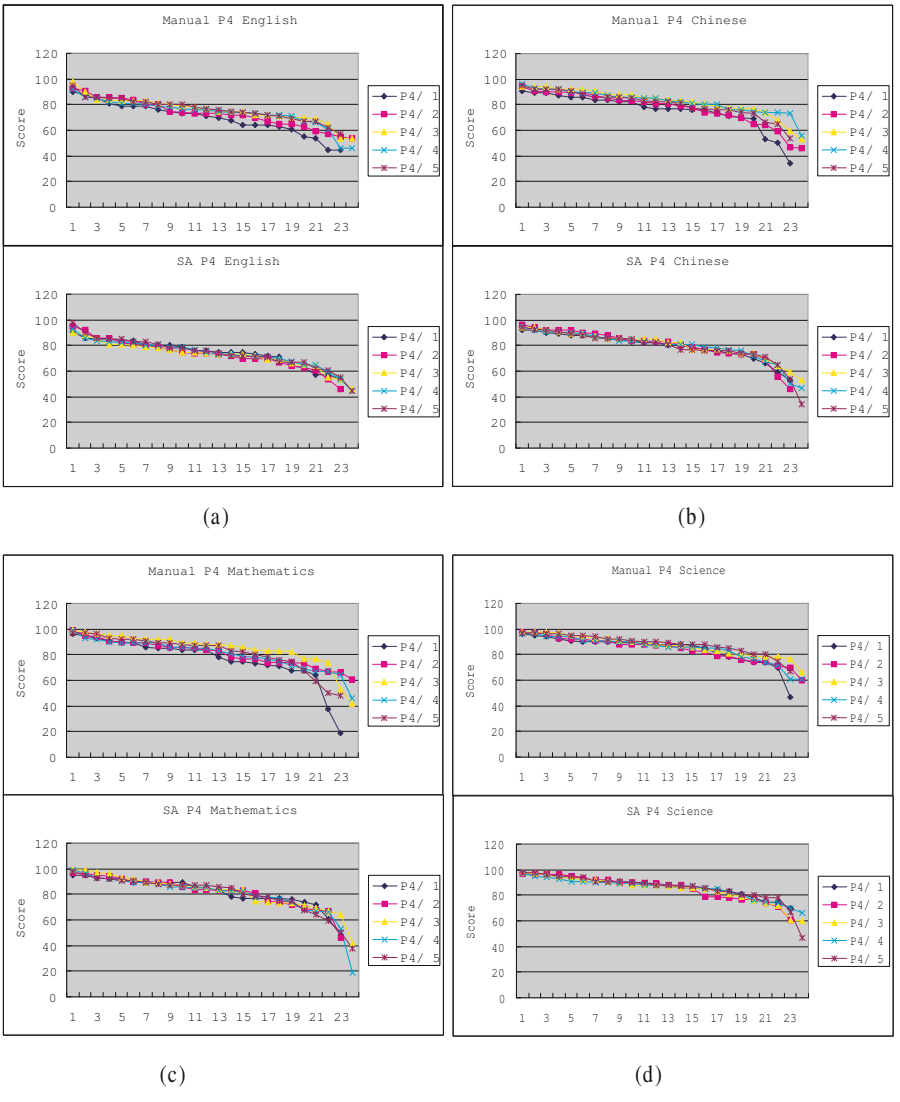


Fig. 5. Distribution curves of English, Chinese, Mathematics and Science scores for all classes in $P4$

SA generated more ideal class sizes by comparing values under heading $M + F$. Fig. 2 shows the values of *Max-Min* (see last row of Fig. 1) for all grades. The smaller numbers under the SA block suggest that SA solutions better satisfy the second objective previously described.

For classes in $P2$, the average scores for each subject were calculated and are presented in Fig. 3. The results show that all students in all classes have nearly equal average scores for each subject, and both the standard deviation and *Max-Min* are very close to zero. The manual results, again, do not match the SA results. Next, we calculated *Max-Min* (see the last row in Fig. 3) for all grades and presented them in Fig. 4. After comparing with the manual results, we observe that SA again fared better in terms of achieving close to average scores of each subject for all classes.

To examine the similarities of the score distribution curves, we used results for $P4$ as an example. Fig. 5 graphically illustrates all distribution curves for $P4$, each of which was constructed by the following: (1) in descending order, sort students in class k according to their subject j's scores; (2) plot the points in a xy-plane with order numbers of students on x-axis and student scores on y-axis. For each subject, the score distribution curves generated by SA are almost the same whereas the curves from the manual operation show a larger degree of dissimilarity.

5 Conclusion

In this paper, we introduced SPS, a partitioning system that our team developed for the Singapore International School Hong Kong. The problem that we addressed stemmed from the real needs and concerns of this school and proved to be strongly NP-hard. We devised a simulated annealing algorithm to solve this problem and the performance of SA in terms of all objectives outperformed that of the manual partitioning process according to the results based on SISHK's data set from the previous academic year.

References

1. Wheelock, A.: Crossing the Tracks: How "Untracking" Can Save America's Schools. The New Press, New York (1992)
2. Garey, M.R., Johnson, D.S.: Computers and Intractibility: A guide to the theory of NP-completeness. W. H. Freeman and Company, San Francisco (1979)
3. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science 220(4598), 671–680 (1983)

Laban-Based Motion Rendering for Emotional Expression of Human Form Robots

Megumi Masuda, Shohei Kato, and Hidenori Itoh

Dept. of Computer Science and Engineering, Graduate School of Engineering,
Nagoya Institute of Technology,
Gokiso-cho Showa-ku Nagoya 466-8555 Japan
{masuda,shohey}@juno.ics.nitech.ac.jp

Abstract. A method modifying arbitrary movement of human form robot (HFR) on the basis of previous work for adding target emotion was developed. The additional emotions are pleasure, anger, sadness or relaxation. This method not only keeps the user interested, but it also makes the user perceive the robot's emotions and form an attachment to the robot more easily. An experiment was conducted using an HFR to test how well our system adds emotion to arbitrary movements. Three basic movements were prepared. Because our aim is to add emotion to an arbitrary movement, the three basic movements were designed to give different impressions. The average of the success rates for adding the target emotion was over 80%. This suggests that our method succeeded in adding the target emotions to arbitrary movements.

1 Introduction

We believe that communication robots will take an active part in our daily lives in the near future. There are many studies about communication robots (e.g., [1], [2]). Moreover, some communication robots are already in use.

Expression of emotions by robots is essential for robot-human communication (RHC). There are two methods for a robot to express an emotional state; one uses verbal information (e.g., [3], [4], [5]) and the other uses nonverbal information (e.g., [6], [7], [8]). Each has advantages and disadvantages in communicative competence and cost performance.

In this paper, whole-body expression as nonverbal information is considered as an expression of emotion (e.g., [9], [10], [11]). Nonverbal information is indispensable for social interaction [12]. Moreover, even with verbal information, bodily expression is important for understanding what is expressed and said [13]. That is appropriate whole-body expression is indispensable for effective RHC.

Creators of a communication robot design the robot's motions carefully with consideration of the above facts. However, preset whole-body movements get old fast though they achieve them best, because those movements have little variation. Therefore, our aim is to create a motion-rendering system that adds emotion to basic movements by modifying the movements. The system can modify the preset movement to reflect the robot's emotional condition. The system

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 49–60, 2010.

not only keeps the user interested, but it also makes the user perceive the robot's emotions and form an attachment to the robot more easily.

There are some studies of selecting behavior suitable for situation (e.g., [14], [15]). Our system will make robot more natural by joining hands with these studies.

Amaya et al. [16] introduced a model to generate emotional animation from neutral human motion. They deal with two emotions (anger and sadness) and two motion features (timing and range) In contrast, we deal with pleasure, anger, sadness and relaxation as emotion. Furthermore, we use six motion features based on Laban movement analysis. Furthermore, a signature of our study is not to pay attention to the emotion expressed by human but to pay attention to the emotion estimated by human. The reason is that our aim is to **express** emotion **for human**.

First, we introduce Laban movement analysis (LMA) and Laban feature value set, which we defined. Next, we describe the method for adding the target emotion to an arbitrary movement on the basis of an examination of LMA. Finally, an experiment is conducted to confirm the usefulness of our method. Success rates of adding the target emotion was over 80%. This suggests that our method succeeded in adding the target emotions.

2 Correlation between Laban Feature and Emotion Expressed by the Movement

Previously, we proposed a set of motion feature values, called the Laban's feature value set, on the basis of Laban movement analysis (LMA). We explained how to distill Laban's feature values from the movement of a human form robot (HFR) and examined the correlation between a robot's whole-body movements and its emotions, estimated by an observer [17]. In this paper, the emotions used are pleasure, anger, sadness and relaxation. These are considered the main emotions in Japan. In this section, we discuss the features of LMA, and the correlations between a robot's whole-body movements and its expressed emotions.

2.1 Laban Movement Analysis

Laban movement analysis (LMA) [18] is a well known theory in dance for observing, describing, notating, and interpreting human movement. It was developed by a German named Rudolf von Laban (1879 to 1958), who is widely regarded as a pioneer of European modern dance and a theorist of movement education. It has succeeded Darwin's movement theory [19], which focuses on the structure of an animal's bodily expression. The general framework was described in 1980 by Irmgard Bartenieff, a scholar of Rudolf von Laban [20]. In neuroscience, the usefulness of LMA to describe certain effects on the movement of animals and humans [21] has been investigated. Laban's theory is well suited for science and engineering, because it is mathematical and specific. The theory of LMA consists of several major components.

Table 1. Laban's features

	Large	Small
Space	the movement directions are one-sided	the movement directions are different
Time	whole-body movement is quick	whole-body movement is slow
Weight	whole-body movement is strong	whole-body movement is weak
Inclination	the whole body is biased forward	the whole body is biased backward
Height	the whole body is biased upward	the whole body is biased downward
Area	the range of whole body is large	the range of whole body is confined

2.2 Laban's Features

The six main features of LMA are *Space*, *Time*, *Weight*, *Inclination*, *Height* and *Area*. We define Laban's features as follows and Table 1.

- ***Space*** represents the bias of whole-body movement.
- ***Time*** represents the quickness of whole-body movement.
- ***Weight*** represents the powerfulness of whole-body movement.
- ***Inclination*** represents the bias for forward of posture.
- ***Height*** represents the straightness of posture.
- ***Area*** represents the movement range of the HFR's body.

The values of Laban features are distilled at every unit timepoint t. The methods from which we distilled our Laban's features were described concretely in [17].

2.3 Object of Laban Feature Value Set

We use our Laban's feature value set on an HFR. The HFR has human-like degrees-of-freedom and extremities. Humanoid is a robot classification, which resembles HFR. However, I use the word "HFR" because "Humanoid" sometimes refers to robots that have sensors or/and some human-like abilities. I use HFR to mean a robot that has extremities, a head, and human-like degrees-of-freedom.

The reasons an HFR (a robot with a human-like figure) was selected are

- It reduces the user's fear and discomfort around the robot.
- It allows the user to easily understand the emotions expressed by the robot's whole-body movements.
- It allows the user to empathize with the robot.

The experiment was conducted not with a computer agent but with a specific HFR. In this paper, the HFR, KHR-2HV (Degree-of-freedom = 17, height = 353 mm), was used as the object.

2.4 Related Works

There are several studies about distilling Laban features.

The robot used by Nakata et al. [22] had three joints and moves on wheels. In contrast, we proposed the method for distilling a Laban feature value set on

a robot that is an HFR with much more joints. An HFR is the type of robot best adapted to mental-like interaction; therefore, HFRs are expected to interact with humans in various situations in the future.

Though the main object of our Laban's feature value set is HFR, our Laban's feature value set is applicable to robots besides HFRs with minor modifications. For example, if the center bottom is considered as the center of the support foot, then our Laban feature values can be distilled from a robot that moves on wheels.

Maeda et al. [9] also studied emotion detection from body movement. They used images of humans and robots. A weakness of their Laban's feature values is a lack of information retrieved from the images. Therefore, our Laban's feature value set is sensitive to whole-body movements. Moreover, we considered body movements by a real robot, because studies (e.g., [23], [24], [25]) report that a robot agent can create more positive impressions than a virtual agent.

In addition, we generated emotion estimation equations, which use our Laban feature values. Their accuracy rates were more than 85%. This suggests that our Laban feature value set is appropriate to use to distill feature values from the whole-body movements.

2.5 Correlation between Laban Features and Expressed Emotion

The correlations between a robot's whole-body movements and the emotions expressed by the movements were examined [17] and the results are in Table 2. The strength values of emotion expressed by the movements are decided based on humans estimation in a pilot experiment. Light gray represents a positive correlation (significance level is over 1%), and dark gray represents a negative correlation (significance level is over 1%).

This table suggests the following:

- Pleasure correlates with quickness and powerfulness. Moreover, pleasure correlates a bias for backwards posture, the straightness of posture, and the movement range of body.
- Anger correlates with quickness and powerfulness. Moreover, anger also correlates with movement with moves in different directions, and the movement range of body.
- Sadness correlates with slowness and weakness. Moreover, sadness correlates with a bias for forward posture, low posture, and narrowness of the body.
- Relaxation correlates with slowness and weakness. Moreover, Relaxation correlates with movements in the same direction, the bias for backwards posture, and the straightness of posture.

3 Method for Adding Emotion to an Arbitrary Basic Movement

In this section, we describe the method for adding the target emotion to the arbitrary movements. We created a motion-rendering system with the consideration

Table 2. Correlations between Laban Feature and Expressed Emotion

	Spa	*Tim*	*Wei*	*Inc*	*Hei*	*Are*
Pleasure	-0.04	0.45	0.46	-0.27	0.33	0.36
Anger	-0.21	0.30	0.33	0.01	-0.02	0.20
Sadness	0.03	-0.38	-0.42	0.47	-0.51	-0.39
Relaxation	0.16	-0.15	-0.12	-0.37	0.36	0.01

that movement can be emotive if it is processed on the basis of the correlation between a robot's whole-body movements and its emotions (Table 2). In this paper, arbitrary movements are limited to movements that do not use the feet to reduce the risk of falling.

3.1 Method for Adding Emotion to Arbitrary Basic Movements

Our method, whose aim is to add emotion to an basic movement, processes the basic movement to change the Laban feature values on the basis of the correlation between a robot's whole-body movements and its emotions (Table 2).

- ***Space*** represents the bias of whole-body movement.
  The bias of whole-body movement is related to the motion direction of the extremities and the direction of the face. It is difficult to change the motion directions of extremities to change the bias without information about the basic movement, because we want the system to be adaptable to arbitrary movements. Therefore, the system modifies the direction of the face. If the direction of the face is near the average direction of the motion directions of extremities, we consider the directions of movement to be one-sided.
- ***Time*** represents the quickness of whole-body movement.
  The quickness is related to the angle velocities. So, the system modifies movement to be quicker. There are two ways to perform quick movement. Former is to rotate the same angle in a shorter time. The latter is to rotate a greater angle in the same amount of time. We adopt the former manner because the system should be applicable to as many basic movements as possible.
- ***Weight*** represents the powerfulness of whole-body movement.
  Powerfulness is related to angle accelerations. It is difficult to change angle acceleration without information about the basic movement because we want the system to be adaptable to arbitrary movements. With respect to adding the emotion to movement, fortunately, it is a similar modification between powerfulness and quickness. Acceleration is changed sufficiently by changing quickness.
- ***Inclination*** represents the bias for forward posture.
  The bias for forward posture is related to the center gravity of the body. The system makes the object bend forward by changing the angle of the waist
- ***Height*** represents the straightness of posture.
  The straightness of the posture is related to the center gravity of the body. The system raises both hands of the HFR.

– ***Area*** represents the movement range of the body.
The movement range of the body is related to the quadrilateral area, which is made by the four extremity points of the extremities, on the horizontal plane. The system makes both hands nearly horizontal.

3.2 Concrete Method for Adding Emotion to Arbitrary Basic Movement

We describe the concrete method for adding emotions to arbitrary basic movements. The foregoing human form robot (HFR), KHR-2HV, was used as the agent.Figure 2 is a photograph of the KHR-2HV. Figure 1 is a link structure of the HFR and information about the method.

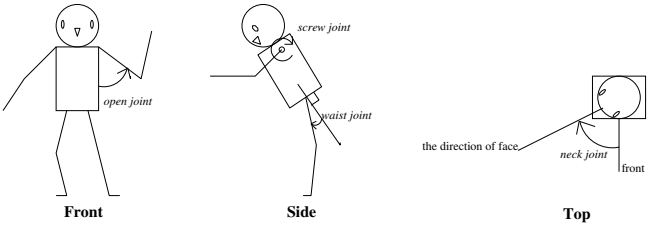


Fig. 1. Link Structure of HFR and Information

– ***Space***
We modify the direction of the face to add emotion as follows.

$$\theta_{head}(t) \rightarrow \theta_{head}(t) + (\theta_{max}(t) - \theta_{head}(t)) \times strength \times emo \qquad (1)$$

θ_{head} is the angle of the face ($neck\ joint$) and θ_{max} is the average of all motion directions of extremities. The emo is the correlation coefficient between anger or relaxation and $Space$. The $strength$ is the strength of the emotion the user inputs.

– ***Time***
We shorten the time necessary to rotate a certain angle to add emotion as follows.

$$time\_interval(t) \rightarrow time\_interval(t)(1 - a * strength) \qquad (2)$$

The $time - interval$ is the time necessary to rotate a certain angle. The $strength$ is the strength of the emotion the user inputs. The a is the weight coefficient.

– ***Weight***
Powerfulness changes at linearly with $Time$, because powerfulness is related to the angle acceleration

$$\ddot{\theta}(t) \rightarrow \frac{\dot{\theta}(t+1)}{(1 - a \times strength)} - \frac{\dot{\theta}(t)}{(1 - a \times strength)} = \frac{\ddot{\theta}(t)}{(1 - a \times strength)} \qquad (3)$$

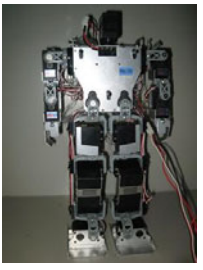


Fig. 2. KHR-2HV

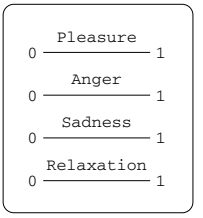


Fig. 3. Questionnaire

– ***Inclination***
We modify the angle of the waist to add emotion as follows.

$$\theta_{waist}(t) \rightarrow \theta_{waist}(t) + b \times strength \times emo \tag{4}$$

θ_{waist} is the angle of the waist (*waist joint*). A positive direction of θ_{waist} is forward. The *emo* is the correlation coefficient between pleasure, sadness, or relaxation and *Inclination*. The *strength* is the strength of emotion the user inputs. The b is the weight coefficient.

– ***Height***
We modify the angle of the shoulder joint to add emotion as follows.

$$\theta_{shoulder}(t) \rightarrow \theta_{shoulder}(t) + c \times strength \times emo \tag{5}$$

$\theta_{shoulder}$ is the angle of the shoulder (*screw joint*). When it is bigger, the arms are turned more towards the top. The *emo* is the correlation coefficient between pleasure, sadness, or relaxation and *Height*. The *strength* is the strength of emotion the user inputs.

– ***Area***
We modify the angle of the shoulder joint to add emotion as follows.

$$\theta_{shoulder2}(t) \rightarrow \theta_{shoulder2}(t) - (180 - \theta_{shoulder2}(t))(strength \times emo) \quad (when\ \theta_{shoulder2}(t)\ is\ over\ 90) \tag{6}$$

$$\theta_{shoulder2}(t) \rightarrow \theta_{shoulder2}(t) + \theta_{shoulder2}(t) strength \times emo \quad (when\ \theta_{shoulder2}(t)\ is\ under\ 90) \tag{7}$$

$\theta_{shoulder2}$ is the angle of the shoulder (*open joint*). $\theta_{shoulder2}$ is 90 when the arm turns to horizontal. The *emo* is the correlation coefficient between pleasure, anger, and sadness and *Area*. The *strength* is the strength of emotion the user inputs.

4 Impression Assessment

We conduct an experiment to test the usefulness of our method. The experiments is an experiment to see if people could identify the emotions we attempted to add to the robot's movements.

There were fifteen subjects between the ages of 20 and 40. They observed the human form robot's (HFR's) whole-body movements for about eight seconds and estimated its emotion. Three basic movements were prepared. There were eight processed movements for the each of the basic movements. Additional emotions are pleasure, anger, sadness, and relaxation. The strength of each additional emotion was weak (emotional strength of 0.5) or strong (emotional strength of 1.0).

4.1 Experimental Procedure

We explained the experimental procedure to the subjects and conducted the experiment. The questionnaires used for the experiment is shown in Figures 3. The procedure of the experiment is as follows.

1. Subjects watch the basic movement of the HFR. The subjects estimate and mark how strongly they think each emotion is expressed through the basic movement. They marked the segment in the questionnaire in Figure 3 to answer. The more strongly the subject perceives that the robot expresses the target emotion, the closer the mark should be to 1. The less emotion the subject perceives, the closer the mark should be to 0. If the subject marks 0, it means the subject estimates that the HFR does not feel the emotion. Homogeneous transformation following the marking was done to quantify the estimation. The quantified estimations are called "values of estimated emotion". The averages of all values of estimated emotion are called the "average of subject's estimations".
2. The subjects watch the processed movements. The subjects estimate and mark how strongly they think each emotion is expressed through the processed movement. They marked the segment in the questionnaire sheets in Figure 3 to answer. The answers are treated the same as the answers of 1.
3. The subjects watch the basic movement again.
4. Two and 3 are repeated until processed movements run out.

Moreover, we present any movement again if the subject requests it.

The above experiment was conducted three times with different basic movements.

4.2 The Basic Movement

In this paper, the basic movement used is a beckoning motion. There are three reasons we selected the beckoning motion. First, there are no presupposed emotions expressed by a beckoning motion. Next, there are situations in which people beckon with emotion. Finally, there is no risk of falling caused by the beckoning motions.

Because our aim is to add emotion to the arbitrary movement, the three basic movements were made as movements giving different impressions. In the three beckoning movements, two of the movements are beckoning with both hands and one movement is beckoning with one hand (Motion 1). One of movement

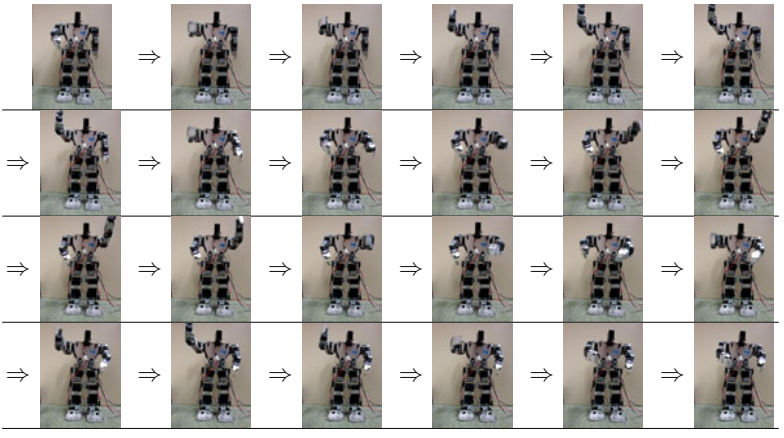


Fig. 4. Sample of Motion

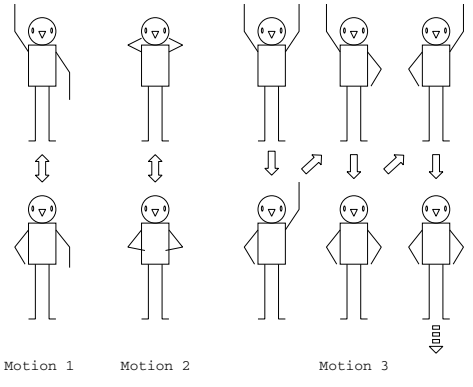


Fig. 5. Beckoning Motions

beckoning with both hands is symmetrical (Motion 2) and the other is not (Motion 3). Figure 5 shows these motions. Continuous snapshots of the movement of Motion 2 is shown in Figure 4.

5 The Rate of Success

The averages of subjects' estimations are shown in Table 3. The table suggests that the three basic movements give different impressions.

5.1 Wilcoxon Signed-Rank Test

We conducted Wilcoxon signed-rank test to confirm that target emotions are added. The result of Wilcoxon signed-rank test are shown in Table 4. The success rates of

Table 3. Averages of Subjects' Estimations of Basic Movements

	Motion 1	Motion 2	Motion 3
Pleasure	32	11	71
Anger	29	11	24
Sadness	15	28	4
Relaxation	12	20	12

Table 4. The Result of Wilcoxon Signed-rank Test

		Motion 1	Motion 2	Motion 3
Pleasure	Weak	**		
	Strong	**	**	*
Anger	Weak		**	
	Strong	*	**	**
Sadness	Weak	*	**	**
	Strong	**	**	**
Relaxation	Weak	**	*	*
	Strong	**	**	*
	*	87.5%	87.5%	75.0%
	**	62.5%	75.0%	37.5%

* : significance level over 5%
** : significance level over 1%

Table 5. Success Rates of Adding Target Emotion

	Pleasure	Anger	Sadness	Relaxation	Success Rate
significance level over 5%	66.7%	66.7%	100.0%	100.0%	83.3%
significance level over 1%	50.0%	50.0%	83.3%	50.0%	58.3%

adding target emotion are shown in Table 5. In this paper, successful motion rendering is defined as satisfying the condition that processed movment is evaluated as significantly higher than its basic movement by subjects' emotion estimation.

First, we consider the result of adding the target emotion **to each basic movement**. All basic movements had high rates of success. This suggests that our method can add emotions to arbitrary movements. The rate of success for adding the target emotion to Motion 3 is comparatively low. The rate of success for adding pleasure or relaxation to Motion 3 is low. We think the reason is that the strength of pleasure for Motion 3 is high (see Table 3). It is probable that it is difficult to add pleasure or relaxation because the basic movement already expresses strong happiness.

Next, we consider the result of **adding each target emotion**. All emotions had a high rate of success. This suggests that our method can add the target emotions to the movements. The rate of success for adding sadness or relaxation is especially high. Meanwile the rate of success for adding pleasure or anger is comparatively low. The low success rate of adding pleasure is related to the low success rate of adding pleasure to Motion 3. We think the reason of low success rate of adding anger is that there were few anger movements in previous experiments. Therefore, it is probable that the motion features of anger were not distilled sufficiently.

6 Conclusion

We proposed a set of motion feature values, called the Laban's feature value set, on the basis of Laban movement analysis (LMA). We explained Laban's feature

value set distilled from the movement of a human form robot (HFR) and examined the correlation between a robot's whole-body movements and its emotions as estimated by an observer. We developed a method for adding a target emotion to an arbitrary movement on the basis of examination of Laban movement analysis. We described a concrete method for modifying arbitrary basic movements of a certain human form robot to express emotions. We conducted an experiment to test if our motion-rendering system could add the target emotion to the arbitrary basic movement. The results suggest that our method added the emotion to the arbitrary movements.

There are some challenges. In this experiment, there were only three basic movements. In the future, we want to experiment with more movements to confirm the usefulness of our method for all basic movements. Besides, we want to study human's emotion estimation from movie using Laban feature value set. It will make a contribution to robot-human bidirectional communication.

References

1. Bennewitz, M., Faber, F., Joho, D., Behnke, S.: Fritz – A Humanoid Communication Robot. In: Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 1072–1077 (2007)
2. Mitsunaga, N., Miyashita, T., Ishiguro, H., Kogure, K., Hagita, N.: Robovie-IV: A Communication Robot Interacting with People Daily in an Office. In: IROS, pp. 5066–5072. IEEE, Los Alamitos (2006)
3. Breazeal, C.: Emotion and Sociable Humanoid Robots. International Journal of Human-Computer Studies 59, 119–155 (2002)
4. Hara, I., Asano, F., Asoh, H., Ogata, J., Ichimura, N., Kawai, Y., Kanehiro, F., Hirukawa, H., Yamamoto, K.: Robust Speech Interface Based on Audio and Video Information Fusion for Humanoid HRP-2. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004), pp. 2404–2410 (2004)
5. Yamamoto, S., Nakadai, K., Nakano, M., Tsujino, H., Valin, J.M., Komatani, K., Ogata, T., Okuno, H.G.: Real-Time Robot Audition System That Recognizes Simultaneous Speech in the Real World. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006), pp. 5333–5338 (2006)
6. Brooks, A.G., Arkin, R.C.: Behavioral overlays for non-verbal communication expression on a humanoid robot. Autonomous Robots 22, 55–74 (2007)
7. Itoh, C., Kato, S., Itoh, H.: A Characterization of Sensitivity Communication Robots Based on Mood Transition. In: Ho, T.-B., Zhou, Z.-H. (eds.) PRICAI 2008. LNCS (LNAI), vol. 5351, pp. 959–964. Springer, Heidelberg (2008)
8. Breazeal, C.: Emotive qualities in robot speech. In: Proc. IROS 2001 (2001)
9. Maeda, Y., Tanabe, N.: Basic Study on Interactive Emotional Communication by Pet-type Robot. Transactions of the Society of Instrument and Control Engineers 42, 359–366 (2006) (in Japanese)
10. Hattori, M., Nakabo, Y., Tadokoro, S., Takamori, T., Yamada, K.: An analysis of the Bunraku puppet's motions based on the phase correspondence of the puppet's motions axis-for the generation of humanoid robots motions with fertile emotions. In: IEEE International Conference on Systems, Man, and Cybernetics, pp. 1041–1046 (1999)

11. Mizoguchi, H., Sato, T., Takagi, K., Nakao, M., Hatamura, Y.: Realization of Expressive Mobile Robot. In: IEEE International Conference on Robotics and Automation (ICRA 1997), pp. 581–586 (1997)
12. Fong, T., Nourbakhsh, I., Dautenhahn, K.: A survey of socially interactive robots. Robotics and Autonomous Systems 42, 143–166 (2003)
13. Rogers, W.T.: The Contribution of Kinetic Illustrators towards the Comprehension of Verbal Behavior within Utterances. Human Communication Research 5, 54–62 (2006)
14. Sawada, T., Takagi, T., Fujita, M.: Behavior selection and motion modulation in emotionally grounded architecture for QRIO SDR-4X II. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, vol. 3, pp. 2514–2519 (2004)
15. Kim, Y., Kim, Y., Kim, J.: Behavior selection and learning for synthetic character. In: Proc. of the IEEE Congress on Evolutionary Computation, pp. 898–903 (2004)
16. Amaya, K., Bruderlin, A., Calvert, T.: Emotion from motion. In: Proceedings of the Conference on Graphics Interface 1996, Canadian Information Processing Society, pp. 222–229 (1996)
17. Masuda, M., Kato, S., Itoh, H.: Emotion Detection from Body Motion of Human Form Robot Based on Laban Movement Analysis. In: Yang, J.-J., Yokoo, M., Ito, T., Jin, Z., Scerri, P. (eds.) PRIMA 2009. LNCS, vol. 5925, pp. 322–334. Springer, Heidelberg (2009)
18. Laban, R.V.: Mastery of Movement. Princeton Book Co. Pub. (1988)
19. Darwin, C.: On the Expression of the Emotions in Man and Animals. John Murray, London (1872)
20. Bartenieff, I., Lewis, D.: Body Movement: Coping with the Environment. Gordon and Breach Science, New York (1980)
21. Foroud, A., Whishaw, I.Q.: Changes in the kinematic structure and non-kinematic features of movements during skilled reaching after stroke: A laban movement analysis in two case sturies. Journal of Neuroscience Methods 158, 137–149 (2006)
22. Nakata, T., Mori, T., Sato, T.: Analysis of Impression of Robot Bodily Expression. Journal of Robotics and Mechatronics 14, 27–36 (2002)
23. Wainer, J., Feil-Seifer, D.J., Shell, D.A., Mataric, M.J.: Embodiment and Human-Robot Interaction. In: 16th IEEE International Conference on Robot & Human Interactive Communication, pp. 872–877 (2007)
24. Powers, A., Kiesler, S., Fussell, S., Torrey, C.: Comparing a computer agent with a humanoid robot. In: 2nd ACM/IEEE International Conference on Human-Robot Interaction (HRI 2007), pp. 145–152 (2007)
25. Kidd, C.D., Breazeal, C.: Effect of a Robot on User Perceptions. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004), pp. 3559–3564 (2004)

Self-supervised Mining of Human Activity from CGM

Nguyen Minh The, Takahiro Kawamura, Hiroyuki Nakagawa, Yasuyuki Tahara, and Akihiko Ohsuga

Graduate School of Information Systems, The University of Electro-Communications, Tokyo, Japan
{minh,kawamura,nakagawa,tahara,akihiko}@ohsuga.is.uec.ac.jp

Abstract. The goal of this paper is to describe a method to automatically extract *all* basic attributes namely *actor*, *action*, *object*, *time* and *location* which belong to an activity, and the *transition* between activities in each sentence retrieved from Japanese CGM (consumer generated media). Previous work had some limitations, such as high setup cost, inability of extracting all attributes, limitation on the types of sentences that can be handled, and insufficient consideration of interdependency among attributes. To resolve these problems, this paper proposes a novel approach that treats the activity extraction as a sequence labeling problem, and automatically makes its own training data. This approach has advantages such as *domain-independence*, *scalability*, and *unnecessary hand-tagged data*. Since it is unnecessary to fix the positions and the number of the attributes in activity sentences, this approach can extract *all* attributes and transitions between activities by making *only a single pass* over its corpus.

1 Introduction

The ability of computers to provide the most suitable information based on users' behaviors is now an important issue in context-aware computing [1], ubiquitous computing [11] and social computing [12]. For example, a service delivers shop information based on the users' next destination [24], a service displays advertisements based on the users' behaviors [16], or an experience-sharing service as shown in Figure 1.

To identify the users' behaviors, it is necessary to understand *how to collect activity data*, *how to express or define each activity* and *its relationships*. It is not practical to define each activity and its relationships in advance, because it not only takes enormous cost, but also cannot deal with unpredictable behaviors. On the other hand, there are some projects that are collecting users' everyday event logs by using sensors installed in mobile phone, GPS or RFID tag [2,3], such as the *My Life Assist Service* [24], and the *Life Log* [4]. From these event logs, they try to extract users' activities, and predict the relationships between them. However, Kawamura et al. [10] indicated some problems in this approach, such as large amount of noisy data in event logs, high computational cost, security and privacy

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 61–72, 2010.

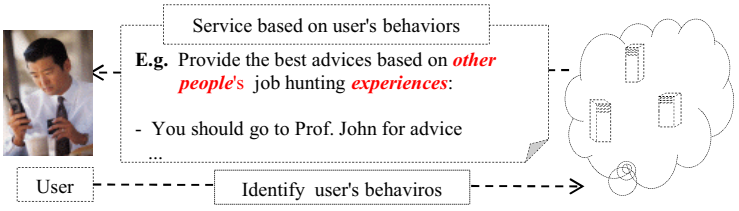


Fig. 1. Experience sharing service

issues. Additionally, if we do not have enough event logs of a large number of different users, it is difficult to discover common or exceptional action patterns.

With such problems discussed above, we try another approach that collects human activity data from CGM. Today, CGM is generated by users posting their activities to Twitter, Facebook, their weblogs or other social media. Thus, it is not difficult to collect activity data of different users from CGM. However, sentences retrieved from CGM have various structures, are complex, are syntactically incorrect. Thus, there are lots of challenges to extract all activity attributes and transitions between activities in these sentences. Few previous works have tried to extract attributes in each sentence retrieved from CGM. These works have some limitations, such as high setup costs because of requiring ontology for each domain [10]. Due to the difficulty of creating suitable patterns, these works are unable to extract all attributes [5,10], limited on the types of sentences that can be handled [5,6], and insufficiently consider interdependency among attributes [5,6].

Since each attribute has interdependent relationships with the other attributes in every activity sentence, we can treat attribute extraction as an *open relation extraction* [13]. In other words, we extract an action and other word phrases that have relationships with this action and describe their activity. In this paper, we propose a novel approach based on the idea of O-CRF [9] that applies self-supervised learning (Self-SL) and uses conditional random fields (CRFs) to the open relation extraction. O-CRF is the state-of-the-art of the open relation extraction from English web pages. Our approach focuses on Japanese CGM, and treats activity extraction as a sequence labeling problem. This approach automatically makes its own training data, and uses CRFs as a learning model. Our proposed architecture consists of two modules: Self-Supervised Learner and Activity Extractor. Given some activity sentences retrieved from the "people" category of Wikipedia, the Learner extracts all attributes and transitions between activities by using deep linguistic parser, and then automatically makes training data. The Learner uses CRFs to make the feature model of these training data. Based on this feature model, the Extractor automatically extracts all attributes and transitions between activities in each sentence retrieved from Japanese CGM.

The main contributions of our approach are summarized as follows:

- It is *domain-independent*, without requiring *any* hand-tagged data.
- It can extract *all* attributes and transitions between activities by making only a *single pass* over its corpus.

– It can handle *all* of the standard sentences in Japanese, and achieves high precision on these sentences.
– It can avoid the privacy problem.

The remainder of this paper is organized as follows. In section 2, we indicate challenges of extracting attributes in more detail. Section 3 explains how our approach makes its own training data, and extracts activity in each sentence retrieved from Japanese CGM. Section 4 reports our experimental results, and discuss how our approach addresses each of the challenges to extract activity attributes. Section 5 considers related work. Section 6 consists of conclusions and some discussions of future work.

2 Challenges

2.1 Activity Attributes Definition

The key elements of an activity are actor, action, and object. To provide suitable information to users, it is important to know *where and when activity happens*. Therefore, in this paper, we define an activity by five basic attributes: actor, action, object, time, and location. We label these attributes as *Who*, *Action*, *What*, *When* and *Where* respectively, and label the transitions between activities as *Next* or *After*. For example, Figure 2 shows the attributes and the transition between activities derived from a Japanese sentence.

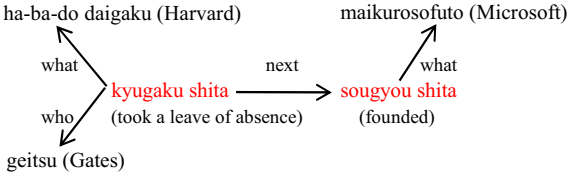


Fig. 2. The attributes and the transition between the activities derived from the activity sentence "*geitsu* ha *ha-ba-do daigaku* wo *kyugaku shi*, *maikurosofuto* wo *sougyou shita*" (Gates took a leave of absence from Harvard, then founded Microsoft).

2.2 Challenges of Extracting Activity Attributes

Extracting activity attributes in sentences retrieved from CGM has many challenges, especially in Japanese. Below, we explain some of them:

1. As shown in Figure 3, O-CRF extracts binary relations in English, and these relations must occur between entity's names within the same sentence [9]. Japanese sentences do not follow this rule, thus we can not directly apply O-CRF for extracting activity attributes in Japanese.
2. In Japanese, there are not word spaces, further word boundaries are not clear. However, previous works in CRFs assume that observation sequence (word) boundaries were fixed. Therefore, a straightforward application of CRFs is impossible.

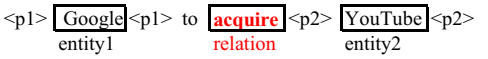


Fig. 3. Limitation of O-CRF

3. Whereas almost typical sentences in English follow the *subject-verb-object* rule [9], Japanese sentences are flexible with many types of structures.
4. Since number and position of attributes are changing in different sentences, it is difficult to create instances or patterns to extract all attributes and transitions between activities.
5. It is not practical to deploy deep linguistic parsers, because of the diversity and the size of the Web corpus [9]. Additionally, sentences retrieved from CGM are often diversified, complex, syntax wrong, and have emoticons. Therefore, the deep linguistic parsers often have errors when parsing these sentences.
6. If extraction method is domain-dependent, then when shifting to a new domain it will require a new specified training examples. And, the extraction process has to be run, and re-run for each domain.

3 Self-supervised Mining of Human Activity

3.1 Activity Extraction with CRFs

CRFs [17] are undirected graphical models for predicting a label sequence to an observed sequence. The idea is to define a conditional probability distribution over label sequences given an observed sequence, rather than a joint distribution over both label and observed sequences. CRFs offers several advantages over hidden Markov models and stochastic grammars, including the ability of relaxing strong independence assumptions made in those models. Additionally, CRFs also avoids the label bias problem, which is a weakness exhibited by maximum entropy Markov models (MEMMs) and other conditional Markov models based on directed graphical models. CRFs achieves high precision on many tasks including text chunking [18], named entity recognition [19], Japanese morphological analysis [20].

By making a first-order Markov assumption that has dependencies between output variables, and arranging variables sequentially in a linear chain, activity extraction can be treated as a sequence labeling problem. Figure 4 shows an example where activity extraction is treated as a sequence labeling problem. Tokens in the surrounding context are labeled using the IOB2 format. B-X means "begin a phrase of type X", I-X means "inside a phrase of type X" and O means "not in a phrase". IOB2 format is widely used for natural language tasks. In this paper, we use CRF++[1] to implement this linear chain CRF.

[1] Available at http://crfpp.sourceforge.net/

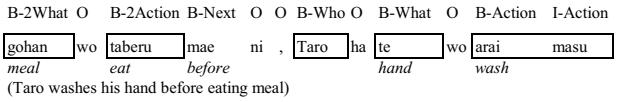


Fig. 4. Activity Extraction as Sequence Labeling

3.2 Proposed Architecture

As shown in Figure 5, the architecture consists of two modules: *Self-Supervised Learner* (I in Figure 5) and *Activity Extractor* (II in Figure 5). Sentences retrieved from the "people" category of Wikipedia are often syntactically correct, activity describable, and easy to parse. Therefore, we parse these sentences to get activity sentences (that describe activities), and then send these activity sentences as sample data to the Learner. The Learner deploys deep linguistic parser to analyze the dependencies between word phrases. Based on the prepared list of Japanese syntax, it selects trustworthy attributes to make training data, and the feature model of these data. The Extractor does *not* deploy deep parser, it bases on this feature model to automatically extract all attributes, and transitions between activities in sentences retrieved from Japanese CGM. Below, we describe each module in more detail.

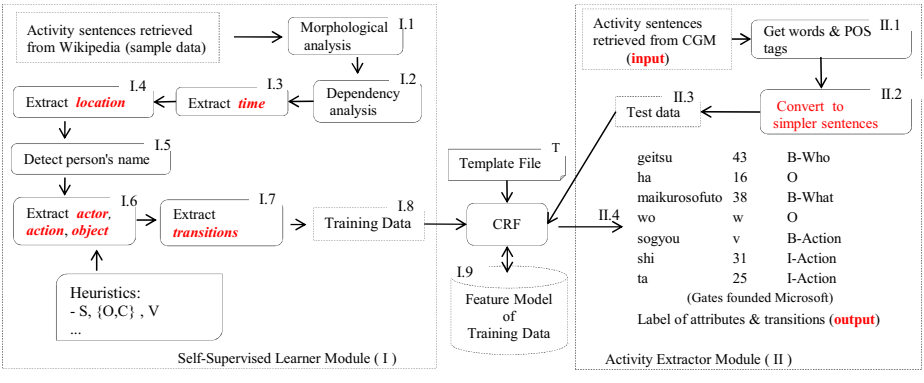


Fig. 5. Proposed Architecture: By using deep linguistic parser, the Learner automatically makes its own training data

3.2.1 Self-supervised Learner Module

We will use the example sentence "geitsu ha maikurosofuto wo sougyou shita" (Gates founded Microsoft) to explain how the Learner works and makes its own training data. As shown in Figure 5, the Learner consists of nine key tasks:

1. By using Mecab[2], it parses the sample data to get words and their POS tags in each sentence (I.1 in Figure 5).

[2] Available at http://mecab.sourceforge.net/

2. By using Cabocha[3], it analyzes the interdependencies among word phrases in each sentence (I.2 in Figure 5). Up to this step, the Learner can detect verb phrase (VP), noun phrase (NP), POS tags, named entity, and the interdependencies among word phrases in each sentence.
3. In addition to the above analytical result, based on the Japanese regular time-expressions such as VP-taato, VP-maeni, toki...etc, the Learner extracts the time of activity and labels it as *When* (I.3 in Figure 5).
4. To improve precision of location extraction, in addition to the above analytical result, the Learner uses the Google map API to extract the location of activity and labels it as *Where* (I.4 in Figure 5).
5. Japanese natural language processing (NLP) tools often have errors when analyzing foreign person name. In this case, the Learner utilizes the "human names" category of Wikipedia to improve precision of person name detection (I.5 in Figure 5).
6. To select trustworthy activity sentences, we prepare a list of all Japanese syntax patterns such as "S, {O, C}, V", "{O, C}, V, S"...etc. Where S means subject, O means object, C means complement, V means verb. Actor, action, object correspond to S, V, O respectively. Based on these syntax patterns, the Learner extracts actor, action, object, and then labels them as *Who*, *Action*, *What* respectively (I.6 in Figure 5).
7. Based on syntax patterns such as V-taato...etc, the Learner extracts the transitions between activities, and labels as *Next* or *After* (I.7 in Figure 5).
8. As shown in Figure 6, training data are automatically created by combining the above results (I.8 in Figure 5).

B-Who	O	B-What	O	B-Action	I-Action
43	16	45	w	v	25
geitsu	ha	maikurosofuto	wo	sougyou	shita
Gates		Microsoft		founded	(Gates founded Microsoft)

Fig. 6. Training data of the example sentence

9. The Leaner uses CRF and template file to automatically generate a set of feature functions (f 1, f 2, ..., f n) as illustrated in Figure 7. The feature model of these training data is created from this set of feature functions (I.9 in Figure 5).

```
f 1 = if (label  = "B-Who" and POS="43")  return 1  else  return 0
....
f n = if (label  = "B-Action" and POS="v")  return 1  else  return 0
```

Fig. 7. Feature functions

[3] Available at http://chasen.org/~taku/software/cabocha/

3.2.2 Activity Extractor Module

We parse Japanese CGM pages to receive activity sentences, and then remove emoticons, and stop words in these sentences. In this case, stop words are the words which do not contain important significance to be used in activity extraction. After this pre-processing, we send activity sentences to the Extractor. As shown in Figure 5, the Extractor consists of four key tasks:

1. The Extractor uses Mecab to get words and their POS tags (II.1 in Figure 5). As shown in Figure 8, in addition to analytical result by Mecab, the Extractor utilizes html tags to detect a long or complex noun phrases.

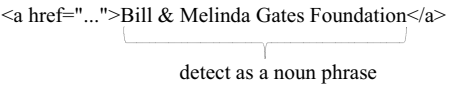


Fig. 8. Using html tags to detect a noun phrase

2. Sentences retrieved from CGM are complex, thus the Extractor converts these sentences to simpler sentences by simplifying noun phrases and verb phrases (II.2 in Figure 5). When converting, it keeps the POS tags of these word phrases.
3. The Extractor makes test data by combining the above results (II.3 in Figure 5). As shown in Figure 9, unlike training data, test data does not have label row. This label row is predicted when testing.

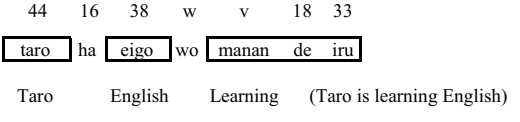


Fig. 9. Test data for the activity sentence *"taro ha eigo wo manan de iru"*

4. Based on the feature model, the Extractor automatically extracts all attributes and transitions between activities in each sentence of the test data (II.4 in Figure 5).

3.2.3 Template File

We use the feature template file to describe features that are used in training and testing (T in Figure 5). The set of features includes words, verbs, part-of-speech (POS) tags and postpositional particles in Japanese. To model long-distance relationships, this paper uses a window size of 7.

4 Evaluation

4.1 Experimental Results

To evaluate the benefits of our approach, we used the set of 533 activity sentences[4] randomly retrieved from Japanese CGM. There are 356 sentences that describe one activity, 177 sentences that describe two activities in this experimental data. Figure 10 shows two sentences which are used for this experiment.

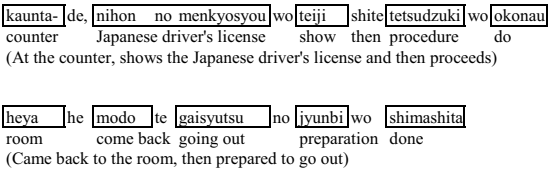


Fig. 10. Two activity sentences in our experimental data

In this experiment, we say an activity extraction is correct when all attributes of this activity are correctly extracted. The precision of each attribute is defined as the number of correctly extracted attributes divided by the total number. Using one PC (CPU: 3.2GHz, RAM: 3.5GB), the Extractor module makes only a single pass over the entire experimental data set, and gets the results[5] as shown in Table 1. This process took only 0.27s.

Table 1. Experimental Results

@	Should be extracted	Correct	Precision (%)
Activity	710	631	88.87
Actor	196	182	92.86
Action	710	693	97.61
Object	509	479	94.11
Time	173	165	95.38
Location	130	120	92.31
Transition	26	22	84.62

4.2 Consideration

The experimental results shown that our approach can automatically extract *all* attributes and transitions between activities in each sentences by making *only a single pass* with high precision. Additionally, our method took only 0.27s, while a widely known deep parser such as Cabocha took over 46.45s for parsing the experimental data (our approach outperforms over 172 times). We consider the experimental results as follows:

[4] Available at http://docs.google.com/View?id=dftc9r33_1077g63vrjc5
[5] Available at http://docs.google.com/View?id=dftc9r33_1078cr9hd3mt

- In activity sentence, action corresponds to a verb phrase. We used Mecab, the parser with high precision of detecting verb phrase. Additionally, we simplified complex verb phrases before testing. These are the reasons of high precision (97.61%) when extracting *action*.
- In Japanese sentence, postpositional particles (wo, ni, he) are often allocated between action and object. In addition to this special feature, we simplify complex noun phrases before testing. So that, our approach achieved high precision (94.11%) of *object* extraction.
- In this experimental data, many actors are people's name. Moreover, in typical Japanese sentence, there are "ha" or "ga" behind subject. Therefore, by using these features, we achieved high precision (92.86%) of *actor* extraction.
- In addition to parsing result of Mecab, we utilize expressions of time in Japanese. So that, our approach achieved high precision (95.38%) of *time* extraction.
- By using Google Map API, we can deal with long or complex addresses. This is reason for high precision (92.31%) of *location* extraction.
- In this experimental data, transitions between activities are explicit. So that, we can extract *transitions* between activities in this data.

Below, we describe how our approach resolves the limitations of the previous works outlined in introduction section.

- It is domain-independent, and automatically creates training data. Therefore, our approach does not take high setup costs.
- By treating activity extraction as a sequence labeling problem, and not fixing the position and number of attributes, our approach is able to extract all attributes in an activity sentence.
- We create training data for all typical sentences. Additionally, we simplify complex sentences before testing. These are reasons for which the Extractor could deal with many type of sentences.
- The feature model contains features of dependencies between attributes in each sentence of training data. Based on these features, the Extractor can consider dependencies between attributes in each sentence of testing data.
- From public CGM, we can collect activity data of many different users. And, this approach can avoid privacy problem.

Our approach addresses each of the challenge indicated in section 2 as follows:

1. It does not fix the position and number of attributes in activity sentences. Additionally, it uses the heuristics (syntax patterns) to select trustworthy training data. We also design the template file to handle multi-attributes in each activity sentence.
2. It uses Mecab and html tags to get word phrases in each sentence.
3. Based on a list of all Japanese syntax, it makes training data for all typical sentences.
4. It treat activity extraction as labeling problem. Additionally, it uses machine learning approach to inference new patterns.
5. The Extractor does not deploy deep linguistic parsers. It deletes emoticons and stop words in each sentences retrieved from CGM. Additionally, the Extractor simplifies complex sentences before testing.

6. The Leaner uses common syntax patterns which do not depend on a specified domain. In other words, the Leaner can make training data for any domain.

However, our approach also has some limitations. Firstly, it only extracts activities that are explicitly described in the sentences. Secondly, it has not yet extracted transitions between activities in document-level. Finally, to handle more complex or incorrect syntax sentences, we need improve our architecture.

4.3 Applying to Other Languages

Our proposed architecture focus on Japanese, but it could be applied to other languages by changing suitable syntax patterns (heuristics) for the Leaner. We should also re-design the template file to utilize special features of the applied language.

4.4 Comparison with O-CRF

Because of the differences in tasks (activity, binary relation) and languages (Japanese, English), it is difficult to compare our approach with O-CRF. We try to compare them according to the some criteria as shown in Table 2.

Table 2. Comparison with O-CRF

	O-CRF	Our Method
Language	English	Japanese
Target data	Binary relation	Human activity
Type of sentences can be handled	S-V-O	{O, C}, V; S, {O, C}, V;... all typical syntax
Relation must occur between entities	YES	NO
Requirement of determining entities before extracting	YES	NO

5 Related Work

There are two fields related to our research: relation extraction (RE) and human activity extraction (AE) from the Web. Below, we discuss the previous researches of each field.

5.1 Relation Extraction

The main researches of RE are DIPRE [14], SnowBall [15], KnowItAll [8], Pasca [7], TextRunner [13], O-CRF [9] (the upgraded version of TextRunner).

DIPRE, SnowBall, KnowItAll, and Pasca use bootstrapping techniques applied for unary or binary RE. Bootstrapping techniques often require a small set of hand-tagged seed instances or a few hand-crafted extraction patterns for each domain. In addition, when creating a new instance or pattern, they could possibly

extract unwanted patterns around the instance to be extracted, which would lead to extract unwanted instance from the unwanted patterns. Moreover, it is difficult to create suitable instances or patterns for extracting the attributes and relationships between activities appeared in sentences retrieved from the Web.

TextRunner is the first Open RE system, it uses self-supervised learning and a Naive Bayes classifier to extract binary relation. Because this classifier predicts the label of a single variable, it is difficult to apply TextRunner to extract all of the basic attributes.

5.2 Human Activity Extraction

Previous works on this field are Perkowitz [5], Kawamura [10] and Kurashima [6]. Perkowitz's approach is a simple keyword matching, so it can only be applied for cases of recipe web pages (such as making tea or coffee). Kawamura's approach requires a product ontology and an action ontology for each domain. So, the precision of this approach depends on these ontologies.

Kurashima used JTAG [21] to deploy a deep linguistic parser to extract action and object. It can only handle a few types of sentences, and is not practical for the diversity and the size of the Web corpus. Additionally, because this approach gets date information from date of weblogs, so it is highly possible that extracted time might not be what activity sentences describe about.

6 Conclusions and Future Work

This paper proposed a novel approach that automatically makes its own training data, and uses CRFs to automatically extract *all* attributes and transitions between activities in each sentence retrieved from Japanese CGM. Without requiring any hand-tagged data, it achieved high precision by making only a *single pass* over its corpus. This paper also explained how our approach resolves the limitations of previous works, and addresses each of the challenges to activity extraction.

We are improving the architecture to handle more complex or syntax incorrect sentences. Based on links between web pages, we will try to extract transitions between activities at the document-level. In the next step, we will use a large data set to evaluate our approach. We are also planning to build a large human activity semantic network based on mining human experiences from the entire CGM corpus.

References

1. Matsuo, Y., Okazaki, N., Izumi, K., Nakamura, Y., Nishimura., T., Hasida, K.: Inferring long-term user properties based on users' location history. In: Proc. IJCAI 2007, pp. 2159–2165 (2007)
2. Pentney, W., Kautz, H., Philipose, M., Popescu, A., Wang, S.: Sensor-Based Understanding of Daily Life via Large-Scale Use of Common Sense. In: Proc. AAAI 2006 (2006)

3. Pentney, W., Philipose, M., Bilmes, J., Kautz, H.: Learning Large Scale Common Sense Models of Everyday Life. In: Proc. AAAI 2007 (2007)
4. KDDI, Corp.: My Life Assist Service (2009), http://www.kddilabs.jp/english/tech/frontier.html
5. Perkowitz, M., Philipose, M., Fishkin, K., Patterson, D.J.: Mining Models of Human Activities from the Web. In: Proc. WWW 2004 (2004)
6. Kurashima, T., Fujimura, K., Okuda, H.: Discovering Association Rules on Experiences from Large-Scale Weblogs Entries. In: ECIR 2009, pp. 546–553 (2009)
7. Pasca, M., Lin, D., Bigham, J., Lifchits, A., Jain, A.: Organizing and Searching the World Wide Web of Facts - Step One: the One-Million Fact Extraction Challenge. In: Proc. AAAI 2006, pp. 1400–1405 (2006)
8. Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D., Yates, A.: Methods for Domain-Independent Information Extraction from the Web: An Experimental Comparison. In: Proc. AAAI 2004 (2004)
9. Banko, M., Etzioni, O.: The Tradeoffs Between Open and Traditional Relation Extraction. In: Proc. ACL 2008 (2008)
10. Kawamura, T., Nguyen, M.T., Ohsuga, A.: Building of Human Activity Correlation Map from Weblogs. In: Proc. ICSOFT (2009)
11. Poslad, S.: Ubiquitous Computing Smart Devices, Environments and Interactions. Wiley, Chichester (2009), ISBN: 978-0-470-03560-3
12. Ozok, A.A., Zaphiris, P.: Online Communities and Social Computing. In: Third International Conference, OCSC 2009, Held as Part of HCI International 2009, San Diego, CA, USA. Springer, Heidelberg (2009), ISBN-10: 3642027733
13. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the Web. In: Proc. IJCAI 2007, pp. 2670–2676 (2007)
14. Brin, S.: Extracting Patterns and Relations from the World Wide Web. In: WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT 1998, Valencia, Spain, pp. 172–183 (1998)
15. Agichtein, E., Gravano, L.: Snowball: Extracting relations from large plain-text collections. In: Proc. ACM DL 2000 (2000)
16. Peppers, D., Rogers, M.: The One to One Future. Broadway Business (1996), ISBN-10: 0385485662
17. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields Probabilistic models for segmenting and labeling sequence data. In: Proc. ICML, pp. 282–289 (2001)
18. Sha, F., Pereira, F.: Shallow parsing with conditional random fields. In: Proc. HLTNAACL, pp. 213–220 (2003)
19. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and Web-enhanced lexicons. In: Proc. CoNLL (2003)
20. Kudo, T., Yamamoto, K., Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphologiaical Analysis. In: IPSJ SIG Notes, pp. 89–96 (2004)
21. Fuchi, T., Takagi, S.: Japanese morphological analyzer using word co-occurence-JTAG. In: Proc. ACL 1998, pp. 409–413 (1998)
22. Kudo, T., Matsumoto, Y.: Japanese Dependency Analysis using Cascaded Chunking. In: Proc. CoNLL 2002, pp. 63–69 (2002)
23. Hiroyuki, Y., Hideyuki, T., Hiromitsu, S.: An individual behavioral pattern to provide ubiquitous service in intelligent space. WSEAS Transactions on Systems, 562–569 (2007)
24. NTTDocomo, Inc.: My Life Assist Service (2009), http://www.igvpj.jp/contents_en/activity09/ms09/list/personal/ntt-docomo-inc-1.html

Data Mining Using an Adaptive HONN Model with Hyperbolic Tangent Neurons

Shuxiang Xu

School of Computing & Information Systems, University of Tasmania, Australia
Shuxiang.Xu@utas.edu.au

Abstract. An Artificial Neural Network (ANN) works by creating connections between different processing elements (artificial neurons). ANNs have been extensively used for Data Mining, which extracts hidden patterns and valuable information from large databases. This paper introduces a new adaptive Higher Order Neural Network (HONN) model and applies it in data mining tasks such as determining breast cancer recurrences and predicting incomes base on census data. An adaptive hyperbolic tangent function is used as the neuron activation function for the new adaptive HONN model. The paper compares the new HONN model against a Multi-Layer Perceptron (MLP) with the sigmoid activation function, an RBF Neural Network with the Gaussian activation function, and a Recurrent Neural Network (RNN) with the sigmoid activation function. Experimental results show that the new adaptive HONN model offers several advantages over conventional ANN models such as better generalisation capabilities as well as abilities in handling missing values in a dataset.

Keywords: neural network, data mining, higher order neural network, adaptive activation function, hyperbolic tangent activation function.

1 Introduction

Data mining is the process of extracting patterns from data. Data mining has become an important tool for transforming data into valuable information. It is commonly used in a wide range of practices such as accounting, marketing, fraud detection, scientific discovery, etc. Traditionally, scientists have manually performed data mining tasks, however, the increasing volume of data in modern society calls for computer-based approaches. The modern technologies of computers together with computer networks have made data collection and organization an easy task. Then the problem is how to retrieve valuable information or patterns from collected data. Data mining is a powerful technology with great potential to help companies survive competition [11, 15].

Data mining is usually supported by the following technologies: massive data collection, powerful multiprocessor computers, and data mining algorithms. The commonly used data mining algorithms include ANNs, Decision Trees, and Rule Induction [5, 15].

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 73–81, 2010.

This paper addresses using ANNs for data mining. ANNs are a natural technology for data mining. ANNs are non-linear models that resemble biological neural networks in structure and learn through training. ANNs present a model based on the massive parallelism and the pattern recognition and prediction abilities of the human brain. ANNs learn from examples in a way similar to how the human brain learns. Then ANNs take complex and noisy data as input and make educated guesses based on what they have learned from the past, like what the human brain does [28].

While conventional ANN models have been able to bring huge profits to many businesses, they suffer from several drawbacks. First, conventional ANN models do not perform well on handling incomplete or noisy data [23, 29, 12]. Next, conventional ANNs can not deal with discontinuities (which contrasts with smoothness: small changes in inputs produce small changes in outputs) in the input training data set [30, 13, 33]. Finally, conventional ANNs lack capabilities in handling complicated business data with high order nonlinearity [13, 32].

To overcome these limitations some researchers have proposed the use of Higher Order Neural Networks (HONNs) [14, 25]. HONNs are networks in which the net input to a computational neuron is a weighted sum of products of its inputs (instead of just a weighted sum of its inputs, as with conventional ANNs). Such neuron is called a Higher-order Processing Unit (HPU) [21]. In [18] a polynomial pipelined HONN is used to predict the exchange rate between the US dollar and three other currencies. The HONN demonstrates more accurate forecasting compared with conventional ANNs. In [31] the functional mapping capability of a HONN model is demonstrated through some well known time series prediction problems, which reveals the advantages of HONN models over conventional ANNs. In [10] a recurrent HONN is developed, with a generalised dynamic back-propagation algorithm derived to show that the proposed HONN not only gives more accurate identification results, but also requires a shorter training time to obtain the desired accuracy. In [6] the superior memory storage capacity of HONN is explored. A non-direct convergence (long-term attraction) analysis of HONN is conducted which leads to promising results. It is also known that HONNs can implement invariant pattern recognition [24, 26].

Adaptive HONNs are HONNs with adaptive activation functions. Such activation functions are adaptive because there are free parameters in the activation functions which can be adjusted (in the same way as connection weights) to adapt to different problems. In [7], an adaptive activation function is built as a piecewise approximation with suitable cubic splines that can have arbitrary shape and allows them to reduce the overall size of the neural networks, trading connection complexity with activation function complexity. In [8], real variables a (gain) and b (slope) in the generalized sigmoid activation function are adjusted during learning process. A comparison with classical ANNs to model static and dynamical systems is reported, showing that an adaptive sigmoid (ie, a sigmoid with free parameters) leads to an improved data modelling.

This paper is organized as follows. Section 2 proposes a new adaptive HONN model with an adaptive generalized hyperbolic tangent activation function. Section 3 presents an adaptive HONN learning algorithm. In Section 4 experiments are conducted to explore the advantages of the new adaptive HONN model (against several

traditional ANN models). Section 5 summarizes this paper and suggests directions for related research in the future.

2 Adaptive HONNs

HONNs were first introduced by [14]. The network structure of a three input second order HONN is shown below:

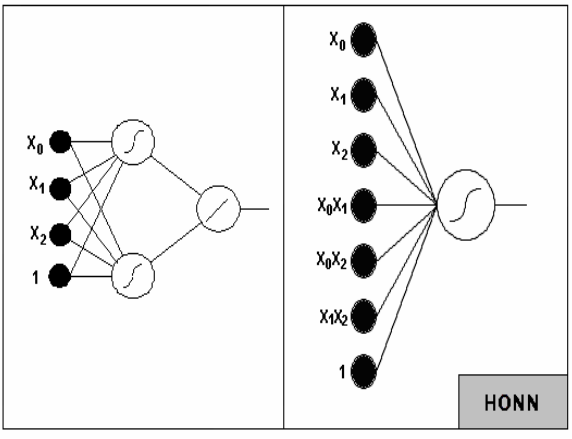


Fig. 1. Left, HONN with three inputs and two hidden nodes; Right, second order HONN with three inputs

Adaptive HONNs are HONNs with adaptive activation functions. The network structure of an adaptive HONN is the same as that of a multi-layer ANN. That is, it consists of an input layer with some input units, an output layer with some output units, and at least one hidden layer consisting of intermediate processing units (see next section on the number of hidden units). We will only use one hidden layer as it has been mathematically proved that ANNs with one hidden layer is a universal approximator [20]. Usually there is no activation function for neurons in the input layer and the output neurons are summing units (linear activation), the activation function in the hidden units is an adaptive one. Our adaptive activation function has been defined as the following (a generalized hyperbolic tangent function):

$$f(x) = \frac{A1 - e^{-B1 \cdot x}}{A2 + e^{-B2 \cdot x}} \tag{2.1}$$

where A1, B1, A2, B2 are real variables which will be adjusted (as well as weights) during training.

Next, an HONN learning algorithm based on an improved steepest descent rule has been developed to adjust the free parameters in the above adaptive activation function (as well as connection weights between neurons). We will see that such approach provides more flexibility and better data mining ability for the new adaptive HONN model.

3 Adaptive HONN Learning Algorithm

We use the following notations:

$I_{i,k}(u)$	the input or internal state of the ith neuron in the kth layer
$w_{i,j,k}$	the weight that connects the jth neuron in layer $k-1$ and the ith neuron in layer k
$O_{i,k}(u)$	the value of output from the ith neuron in layer k
$A1, B1, A2, B2$	adjustable variables in activation function
$\theta_{i,k}$	the threshold value of the ith neuron in the kth layer
$d_j(u)$	the jth desired output value
β	learning rate
m	total number of output layer neurons
l	total number of network layers
η	momentum
r	the iteration number

The input to the ith neuron in the kth layer is:

$$I_{i,k}(u) = \sum_j \left(w_{i,j,k} O_{j,k-1}\right) + \prod_j out_{j,k-1} - \theta_{i,k} \tag{3.1}$$

$$O_{i,k}(u) = \Psi\left(I_{i,k}(u)\right) = \frac{A1_{i,k} - e^{-B1_{i,k} \cdot I_{i,k}(u)}}{A2_{i,k} + e^{-B2_{i,k} \cdot I_{i,k}(u)}} \tag{3.2}$$

To train our neural network an energy function

$$E = \frac{1}{2} \sum_{j=1}^{m} \left(d_j(u) - O_{j,l}(u)\right)^2 \tag{3.3}$$

is adopted, which is the sum of the squared errors between the actual network output and the desired output for all input patterns. In (3.3), m is the total number of output layer neurons, l is the total number of constructed network layers. The aim of learning is undoubtedly to minimize the energy function by adjusting the weights associated with various interconnections, and the variables in the activation function. This can be fulfilled by using a variation of the steepest descent gradient rule expressed as follows:

$$w_{i,j,k}^{(r)} = \eta w_{i,j,k}^{(r-1)} + \beta \frac{\partial E}{\partial w_{i,j,k}} \tag{3.4}$$

$$\theta_{i,k}^{(r)} = \eta\theta_{i,k}^{(r-1)} + \beta\frac{\partial E}{\partial \theta_{i,k}} \tag{3.5}$$

$$A1_{i,k}^{(r)} = \eta A1_{i,k}^{(r-1)} + \beta\frac{\partial E}{\partial A1_{i,k}} \tag{3.6}$$

$$B1_{i,k}^{(r)} = \eta B1_{i,k}^{(r-1)} + \beta\frac{\partial E}{\partial B1_{i,k}} \tag{3.7}$$

$$A2_{i,k}^{(r)} = \eta A2_{i,k}^{(r-1)} + \beta\frac{\partial E}{\partial A2_{i,k}} \tag{3.8}$$

$$B2_{i,k}^{(r)} = \eta B2_{i,k}^{(r-1)} + \beta\frac{\partial E}{\partial B2_{i,k}} \tag{3.9}$$

To derive the gradient information of E with respect to each adjustable parameter in equations (3.4)-(3.9), we define

$$\frac{\partial E}{\partial I_{i,k}(u)} = \zeta_{i,k} \tag{3.10}$$

$$\frac{\partial E}{\partial O_{i,k}(u)} = \xi_{i,k} \tag{3.11}$$

Now, from equations (3.2), (3.3), (3.10) and (3.11), we have the partial derivatives of E with respect to adjustable parameters as follows:

$$\frac{\partial E}{\partial w_{i,j,k}} = \frac{\partial E}{\partial I_{i,k}(u)}\frac{\partial I_{i,k}(u)}{\partial w_{i,j,k}} = \zeta_{i,k} O_{j,k-1}(u) \tag{3.12}$$

$$\frac{\partial E}{\partial \theta_{i,k}} = \frac{\partial E}{\partial I_{i,k}(u)}\frac{\partial I_{i,k}(u)}{\partial \theta_{i,k}} = -\zeta_{i,k} \tag{3.13}$$

$$\frac{\partial E}{\partial A1_{i,k}} = \frac{\partial E}{\partial O_{i,k}}\frac{\partial O_{i,k}}{\partial A1_{i,k}} = \xi_{i,k} e^{-B1_{i,k}\cdot I_{i,k}} \tag{3.14}$$

$$\begin{aligned}\frac{\partial E}{\partial B1_{i,k}} &= \frac{\partial E}{\partial O_{i,k}}\frac{\partial O_{i,k}}{\partial B1_{i,k}} \\ &= -\xi_{i,k}\cdot A1_{i,k}\cdot I_{i,k}\cdot e^{-B1_{i,k}\cdot I_{i,k}}\end{aligned} \tag{3.15}$$

$$\frac{\partial E}{\partial A2_{i,k}} = \frac{\partial E}{\partial O_{i,k}}\frac{\partial O_{i,k}}{\partial A2_{i,k}} = \xi_{i,k}\cdot\frac{1}{1+e^{-B2_{i,k}\cdot I_{i,k}}} \tag{3.16}$$

$$\frac{\partial E}{\partial B2_{i,k}} = \frac{\partial E}{\partial O_{i,k}(u)}\frac{\partial O_{i,k}(u)}{\partial B2_{i,k}} = \xi_{i,k}\cdot\frac{A2_{i,k}\cdot I_{i,k}(u)\cdot e^{-B2_{i,k}\cdot I_{i,k}(u)}}{\left(1+e^{-B2_{i,k}\cdot I_{i,k}(u)}\right)^2} \tag{3.17}$$

And for (3.10) and (3.11) the following equations can be computed:

$$\zeta_{i,k} = \frac{\partial E}{\partial I_{i,k}(u)} = \frac{\partial E}{\partial O_{i,k}(u)}\frac{\partial O_{i,k}(u)}{\partial I_{i,k}(u)} = \xi_{i,k}\cdot\frac{\partial O_{i,k}(u)}{\partial I_{i,k}(u)} \tag{3.18}$$

while

$$\frac{\partial O_{i,k}(u)}{\partial I_{i,k}(u)} = A1_{i,k} \cdot B1_{i,k} \cdot e^{-B1_{i,k} \cdot I_{i,k}(u)} + \frac{A2_{i,k} \cdot B2_{i,k} \cdot e^{-B2_{i,k} \cdot I_{i,k}(u)}}{\left(1 + e^{-B2_{i,k} \cdot I_{i,k}(u)}\right)^2} \tag{3.19}$$

and

$$\xi_{i,k} = \begin{cases} \sum_j \zeta_{j,k+1} w_{j,i,k+1}, & \text{if } 1 \le k < l; \\ O_{i,l}(u) - d_i(u), & \text{if } k = l. \end{cases} \tag{3.20}$$

All the training examples are presented cyclically until all parameters are stabilized, i.e., until the energy function E for the entire training set is acceptably low and the network converges.

4 Experiments

Our first experiment is to use the adaptive HONN to handle the Breast Cancer dataset from the UCI Machine Learning Repository [2]. This dataset is made of 286 instances, with 9 attributes (inputs) and 1 class attribute (output) which shows no-recurrence-event or recurrence-event. The dataset is made of 201 instances of no-recurrence-events and 85 instances of recurrence-events. There are missing attribute values in this dataset. Based on a new approach for determining the optimal number of hidden layer neurons [30], the optimal number of hidden layer neurons for this experiment is n=5.

For this experiment, the data set is divided into a training set made of 75% of the original set and a test set made of 25% of the original set. A validation set is not used because the optimal number of hidden layer neurons has been determined. After the adaptive HONN (with 5 hidden layer units) has been well trained over the training data pairs, it is used to forecast over the test set. The correctness rate reaches 96.6%. To verify that for this example the optimal number of hidden layer neuron is 4, we try to apply the same procedure by setting the number of hidden layer neurons to 4, 6, and 8, which results in correctness rates of 86.6%, 82.2%, and 78.8% on the test set, respectively.

To verify the advantages of the adaptive HONN model we establish the following ANNs for comparison studies: a Multi-Layer Perceptron (MLP) with the sigmoid activation function (and one hidden layer); An RBF Neural Network with the Gaussian activation function (and one hidden layer); and a Recurrent Neural Network (RNN) with the sigmoid activation function (and one hidden layer). With 5 hidden neurons for each of the three ANNs and the same training set, the MLP reaches a correctness rate of 75.2% on the test set, the RBF Network reaches a correctness rate of 79.9%, and the RNN reaches a correctness rate of 81.3%. These results seem to suggest that the adaptive HONN model holds better generalisation capability in handling datasets with missing values. However, the adaptive HONN takes longer time during the training process. Interestingly, in this case, the RNN Network is the fastest learner.

For our second experiment, The Census-Income (KDD) Data Set from the UCI Machine Learning Repository is used [2]. This data set contains weighted census data from the 1994 and 1995 population surveys conducted by the U.S. Census Bureau. There are 299285 instances with 40 attributes (inputs) in this large dataset. There are missing values in it. Based on the new approach [30], the optimal number of hidden layer neurons for this experiment is n=37.

For this experiment, the data set is divided into a training set made of 80% of the original set and a test set made of 20% of the original set. Again a validation set is not used because the optimal number of hidden layer neurons has been determined. After the adaptive HONN (with 37 hidden layer units) has been well trained over the training data pairs, it is used to forecast over the test set. The correctness rate reaches 88.6%. To verify that for this example the optimal number of hidden layer neuron is 37, we try to apply the same procedure by setting the number of hidden layer neurons to 34, 40, and 44, which results in correctness rates of 80.6%, 78.6%, and 72.9% on the test set, respectively.

To verify the advantages of the adaptive HONN model we also establish the following ANNs for comparison studies: a Multi-Layer Perceptron (MLP) with the sigmoid activation function (and one hidden layer); An RBF Neural Network with the Gaussian activation function (and one hidden layer); and a Recurrent Neural Network (RNN) with the sigmoid activation function (and one hidden layer). With 37 hidden neurons for each of the three ANNs and the same training set, the MLP reaches a correctness rate of only 67.4% on the test set, the RBF Network reaches a correctness rate of 72.4%, and the RNN reaches a correctness rate of 71.8%. Again, these results seem to suggest that the adaptive HONN model holds better generalisation capability in handling datasets with missing values. However, the adaptive HONN again takes significantly longer time to learn the input samples (because there are significantly more inputs for this experiment).

5 Conclusions

A new HONN model with an adaptive hyperbolic tangent activation function is introduced in this report, which is applied in data mining tasks such as determining breast cancer recurrences and predicting incomes base on census data. The paper compares the new adaptive HONN model against a MLP with the sigmoid activation function, an RBF Network with the Gaussian activation function, and an RNN with the sigmoid activation function. Experimental results show that the new HONN model seems to offer significant advantages over conventional ANNs such as better generalisation capabilities, and the ability of handling missing values in a dataset. The result that the adaptive HONN model takes longer time to learn the training samples is from the fact that there are significantly more input neurons in a HONN model (compared with models such as MLP). For future work, first, more experiments should be conducted to test the adaptive HONN on other datasets (especially large datasets) to further verify its capabilities. Next, further comparison studies between the adaptive HONN and other ANN approaches should be conducted to demonstrate the advantages of this new approach.

References

1. Ash, T.: Dynamic node creation in backpropagation networks. Connection Science 1(4), 365–375 (1989)
2. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
3. Barron, A.R.: Approximation and Estimation Bounds for Artificial Neural Networks. Machine Learning (14), 115–133 (1994)
4. Barron, A.R., Cover, T.M.: Minimum complexity density estimation. IEEE Transactions on Information Theory 37, 1034–1054 (1991)
5. Bramer, M.: Principles of Data Mining. Springer, Heidelberg (2007)
6. Burshtein, D.: Long-term attraction in higher order neural networks. IEEE Transactions on Neural Networks 9(1), 42–50 (1998)
7. Campolucci, P., Capparelli, F., Guarnieri, S., Piazza, F., Uncini, A.: Neural networks with adaptive spline activation function. In: Proceedings of IEEE MELECON 1996, Bari, Italy, pp. 1442–1445 (1996)
8. Chen, C.T., Chang, W.D.: A feedforward neural network with function shape autotuning. Neural Networks 9(4), 627–641 (1996)
9. Chen, Y.H., Jiang, Y.L., Xu, J.X.: Dynamic properties and a new learning mechanism in higher order neural networks. Neurocomputing 50, 17–30 (2003)
10. Cho, J.S., Kim, Y.W., Park, D.J.: Identification of nonlinear dynamic systems using higher order diagonal recurrent neural network. Electronics Letters 33(25), 2133–2135 (1997)
11. Cios, K.J., Pedrycz, W., Swiniarski, R.W., Kurgan, L.A.: Data Mining: A Knowledge Discovery Approach. Springer, Heidelberg (2007)
12. Dong, G., Pei, J.: Sequence Data Mining (Advances in Database Systems). Springer, Heidelberg (2007)
13. Fulcher, J., Zhang, M., Xu, S.: Application of Higher-Order Neural Networks to Financial Time-Series Prediction. In: Kamruzzaman, J., Begg, R., Sarker, R. (eds.) Artificial Neural Networks in Finance and Manufacturing, ch. V, pp. 80–108. Idea Group Publishing (2006), ISBN: 1591406714
14. Giles, L., Maxwell, T.: Learning Invariance and Generalization in High-Order Neural Networks. Applied Optics 26(23), 4972–4978 (1987)
15. Han, J., Kamber, M.: Data Mining: concepts and techniques. Elsevier, Amsterdam (2006)
16. Haykin, S.: Neural networks: a comprehensive foundation. Prentice Hall, New Jersey (1999)
17. Hirose, Y., Yamashita, I.C., Hijiya, S.: Back-propagation algorithm which varies the number of hidden units. Neural Networks 4 (1991)
18. Hussain, A.J., Knowles, A., Lisboa, P.J.G., El-Deredy, W.: Financial time series prediction using polynomial pipelined neural networks. Expert Systems with Applications 35(3), 1186–1199 (2008)
19. Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V., Saarela, A.: Self organization of a massive document collection. IEEE Trans. Neural Networks 11, 574–585 (2000)
20. Leshno, M., Lin, V.Y., Pinkus, A., Schocken, S.: Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Networks 6, 861–867 (1993)
21. Lippman, R.P.: Pattern classification using neural networks. IEEE Commun. Mag. 27, 47–64 (1989)

22. Masseglia, F., Poncelet, P., Teisseire, M.: Successes and New Directions in Data Mining. Information Science Reference, Publisher (2007)
23. Peng, H., Zhu, S.: Handling of incomplete data sets using ICA and SOM in data mining. Neural Computing & Applications 16(2), 167–172 (2007)
24. Psaltis, D., Park, C.H., Hong, J.: Higher order associative memories and their optical implementations. Neural Networks 1, 149–163 (1988)
25. Redding, N., Kowalczyk, A., Downs, T.: Constructiveq high-order network algorithm that is polynomial time. Neural Networks 6, 997–1010 (1993)
26. Reid, M.B., Spirkovska, L., Ochoa, E.: Simultaneous position, scale, rotation invariant pattern classification using third-order neural networks. Int. J. Neural Networks 1, 154–159 (1989)
27. Rivals, I., Personnaz, L.: A statistical procedure for determining the optimal number of hidden neurons of a neural model. In: Second International Symposium on Neural Computation (NC 2000), Berlin, May 23-26 (2000)
28. Witten, I.H., Frank, E.: Data mining: practical machine learning tools and techniques. Elsevier/Morgan Kaufman, Amsterdam (2005)
29. Wang, S.H.: Application of self-organising maps for data mining with incomplete data sets. Neural Computing & Applications 12(1), 42–48 (2003)
30. Xu, S.: Adaptive Higher Order Neural Network Models and Their Applications in Business. In: Zhang, M. (ed.) Artificial Higher Order Neural Networks for Economics and Business, ch. XIV, pp. 314–329. IGI Global (2008), ISBN: 978-1-59904-897-0
31. Yadav, R.N., Kalra, P.K., John, J.: Time series prediction with single multiplicative neuron model. Applied Soft Computing 7(4), 1157–1163 (2007)
32. Zhang, M., Xu, S.X., Fulcher, J.: ANSER: an Adaptive-Neuron Artificial Neural Network System for Estimating Rainfall Using Satellite Data. International Journal of Computers and Applications 29(3), 215–222 (2007)
33. Zhang, M., Xu, S.X., Fulcher, J.: Neuron-adaptive higher order neural-network models for automated financial data modeling. IEEE Transactions on Neural Networks 13(1), 188–204 (2002)

Business Intelligence for Delinquency Risk Management via Cox Regression

Sung Ho Ha and Eun Kyoung Kwon

School of Business Administration, Kyungpook National University,
1370 Sangyeok-dong, Buk-gu, 702-701 Daegu, Korea
hsh@mail.knu.ac.kr, redviolin12@nate.com

Abstract. The recent economic downturn has made many delinquent credit card customers, which in sequence reduced profit margins as well as sales of the retail companies. This study focuses on customers who have recovered from credit delinquency to analyze repayment patterns of delinquents. A Cox regression model is designed as a credit-predicting model to handle with credit card debtors. The model predicts the expected time for credit recovery from delinquency. The model's prediction accuracy is compared with that of other known algorithms.

Keywords: Delinquency risk, delinquency management, credit recovery, data mining.

1 Introduction

Department stores have been in the center of the distribution industry since the 1960s. Nowadays they are however struggling to escape from the unintended market retrenchment due to the growth of competitive forces, including shopping outlets, catalog shopping, and online home shopping channels. The consumption sentiment decreased by economic depression during the 2000s leads to higher rates of insolvency on the part of customers holding department store credit cards.

In order to cope with this problem, more research needs to focus on credit prediction systems. Most studies have divided customers into two or three groups so far, such as good/bad customers or good/bad/latent bad customers, based upon customer credit. Next, studies focused on the classification of good or bad customers before delinquency [1]. Thus, it is difficult to locate research that considers customer groups who can recover from delinquency to normal credit status. Under these circumstances, credit prediction research should turn its attention to the customer management of delinquents.

This study focuses on customer groups that can recover from a credit delinquent state and devises a credit prediction model for delinquents holding credit cards issued by a department store. This delinquency risk management employs a Cox regression model, a semi-non-parametric technique for a survival analysis. A Cox regression model is used to predict whether credit delinquents will maintain a credit delinquency

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 82–90, 2010.

status and to estimate the time required for credit recovery by identifying the variables that influence credit recovery. This model is expected to increase the efficiency of a department store's delinquency risk management by building differentiated delinquency strategies for each delinquent.

The organization of this study is as follows: Section 2 explains several previous credit prediction models. Section 3 devises a Cox regression model to predict a delinquent's credit state. In Section 4, the model is applied to real-world data, and the performance is compared with that of other prediction methods. Section 5 concludes this study.

2 Literature Review

Credit prediction or credit risk assessment methods are classified into several groups: statistical methods including multivariate discrimination analysis, logistic regression analysis and probit analysis; management sciences including mathematical programming; and data mining methods including decision trees, neural networks, and genetic algorithms [2].

The initial credit risk management systems used statistical methods extensively [3]. Min and Lee [4] proposed a DEA-based approach to credit scoring for the financial data of externally audited 1061 manufacturing firms. Tsaih et al. [5] used a probit regression to develop a credit scoring system for small business loans. Credit prediction systems using survival analysis were proposed more recently. A survival analysis model is a kind of statistical method used to analyze time-zone data about the occurrence of an event [6]. Because the time-zone data usually have some defective or missing parts, a Cox's proportional hazard model has been used in credit prediction [7].

Data mining methods have been used for credit prediction [8]. In particular, studies using machine learning methods have exhibited good performance in credit prediction [9]. A decision tree is said to be the most successful method for credit prediction. The decision tree makes it easier to classify customers into good or bad customers according to their credit status. Chen et al. [10] devised an information granulation approach to tackle the class imbalance problem when detecting credit fraud. A genetic algorithm method is efficient, flexible and can find global optimal solutions from more than two-dimensional search space. It is based on the theory of natural selection and survival of the fittest. Huang et al. [11] proposed two-stage genetic programming to improve the accuracy of credit scoring models.

Artificial neural networks have also shown outstanding results in the credit prediction process. They are appropriate for explaining non-linear models because they do not require a statistical hypothesis. Abdou et al. [12] investigated neural networks in evaluating a bank's personal loan data set in Egypt. West [13] investigated five neural network models of credit scoring to find the most accurate mixture of experts and radial basis function neural network models. Malhotra and Malhotra [14] developed neural networks to classify seven different consumers applying for loans. Hsieh [15] used a self-organizing map neural network to identify profitable groups of customers based on repayment, recency, frequency, and monetary behavioral scores. West et al. [16] employed a multilayer perceptron neural network with cross-validation, bagging,

or boosting strategies for financial decision applications. Yen [17] proposed a method to issue warning signals for potential accounting frauds by using adaptive resonance theory.

Even though there are many credit prediction systems using data mining techniques, it is not easy to tell which one is the best [18]. This is because data mining techniques cannot guarantee the global optimal solution, owing to overfitting, inappropriate learning, or difficulty in building an optimized model. According to [19], single methods have the same limitations. Therefore, it is thought to be more efficient to mix credit prediction models, combining the merits of individual methods to provide the most optimized solutions [20]. Chen et al. [21] applied a hybrid support vector machine (SVM) model, based on a classification and regression tree (CART), multivariate adaptive regression splines (MARS), and a grid search, to a credit card dataset provided by a local Chinese bank. Martens et al. [22] introduced SVM rule extraction techniques that were used for credit risk evaluation.

Chen and Hung [11] made up for the weak points of neural networks by using a genetic algorithm. Lee et al. [23] showed that the performance of a mixed neural network combined with a discriminant analysis was much higher than that of a single neural network. Lee et al. [24] proposed a credit scoring method combining CART and MARS to apply it to a bank credit card data set. Zhu et al. [25] implemented a self-organizing learning array system combined with an information theoretic learning capable of solving financial problems. Laha [26] evaluated credit risk by integrating a fuzzy-rule-based classification and *k*-NN method.

3 Cox Regression Model of Survival Analysis

This study uses a Cox regression model of survival analysis to identify the repayment patterns of credit delinquents. It uses a likelihood function to analyze survival time data [27]. Assuming that a survival time t is equal to or greater than zero ($t \geq 0$), a probability density function for the survival time t is presented in $f(t)$, which means a momentary probability that someone can survive at time t but happen to die during Δt. A cumulative density function $F(t)$ for the time t means that someone has died before or at time t (Equation 1).

$$f(t) = \lim_{\Delta t \to 0} \frac{\Pr\left(t \leq T < t + \Delta t\right)}{\Delta t}, \qquad F(t) = \Pr\left(T \leq t\right) = \int_0^t f(u)du \tag{1}$$

The probability of maintaining a life until time t can be presented as a survival function, $S(t)$, which can be calculated using the following Equation 2. The function $f(t)$ is therefore calculated by multiplying (-1) by the differential coefficient of the survival function.

$$S(t) = \Pr\left(T \geq t\right) = 1 - F(t), \qquad f(t) = -\frac{dS(t)}{dt} \tag{2}$$

This study uses the survival rate and the survival function as "delinquency maintenance rate" and "delinquency maintenance function," respectively. The survival function $S(t)$ presents the probability of maintaining credit delinquency status until time *t*. The density function $f(t)$ represents the moment of status change from credit delinquency status to credit recovery status. A hazard function $h(t)$ calculates the momentary probability of dying at time *t*, given survival through a certain amount of time *t-1*. This study uses the hazard rate and the hazard function as "credit recovery rate" and "credit recovery function," respectively. Thus $h(t)$ represents the momentary probability of credit recovery at *t*, assuming that someone has been in a credit delinquency state until *t-1*. The hazard function can be expressed by using the survival function, as seen in Equation 3.

$$h(t) = \frac{f(t)}{S(t)} = \frac{-\frac{dS(t)}{dt}}{S(t)} = -\frac{d}{dt}\log S(t) \tag{3}$$

where $\log S(t) = -\int_0^t h(u)du$, $S(t) = \exp[-\int_0^t h(u)du] = \exp[-H(t)]$, and $H(t)$ is the cumulative hazard function (CHF).

4 Application and Evaluation

Delinquency data for this study came from a department store in 2008. Delinquent customer information (5365 records), delinquency transaction information, and purchasing transaction information were gathered for the analysis. Customers who had unredeemed debts were not considered at the time of data gathering because the target company retained a credit management policy that considered one-year-overdue debts as irrecoverable debts.

4.1 Discovering of Repayment Patterns

A set of five variables was arranged to build a Cox model: the number of delinquency (del_num), the period of delinquency (del_period), the amount of delinquency (del_amount), the time interval between delinquencies (bet_period), and the number of repayments (ret_num). "Cox Regression Model" in Clementine 12.0 was used to build the survival model to analyze the patterns of repayment for credit recovery. Note that in this study a "survival" means that delinquents remain in delinquency status until time *t*, and a "hazard" means that a delinquent escapes from delinquency at time *t*, but was in the delinquency state at *t-1*.

To record the credit status of a customer, a "status" variable was created and a value of one was assigned if a delinquent had recovered from delinquency. The "status" variable was set up as the target and the period of delinquency was entered as a survival time. "Stepwise estimation method" was chosen to control the criteria for adding and removing variables. The significance thresholds for adding and removing variables were set to 0.05 and 0.1, respectively. The removal criterion was the likelihood- ratio statistic.

The final model included variables such as del_num (p<0.000) and ret_num (p<0.000), and the final regression equation was represented as $0.703 \times del\_num + (-10.103) \times ret\_num$. The model was statistically significant with "-2 log likelihood = 58602.034", "Chi-Square = 1050.206", and "degree of freedom = 2". Table 1 summarizes the baseline hazard function, the cumulative hazard function, and the survival function according to the elapsed time.

Table 1. Baseline hazard, cumulative hazard, and survival functions

Month	Baseline hazard	Cumulative hazard	Survival
2	7.103	0.846	0.429
3	13.432	1.600	0.202
4	19.518	2.324	0.098
5	24.439	2.910	0.054
6	30.736	3.660	0.026
7	36.416	4.337	0.013
8	41.640	4.959	0.007
9	47.008	5.598	0.004
10	54.837	6.531	0.001
11	451.151	53.728	0.000
12	.	.	0.000

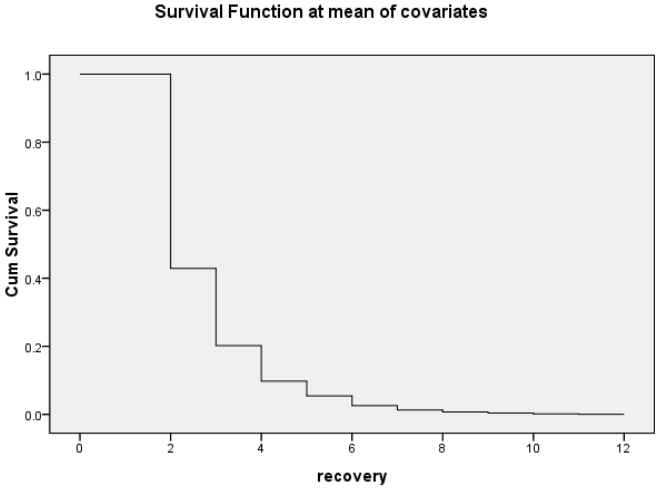


Fig. 1. The survival function of the Cox model

The slope coefficient (β) of del_num is positive, and therefore, as the number of delinquency increases, the amount of time required for credit recovery increases. On the other hand, the slope coefficient of ret_num is negative. Accordingly, the amount of time for credit recovery from delinquency decreases as the number of repayments for credit recovery increases. Fig. 1 shows the resulting graph of a survival function

of the Cox model. In this figure, the cumulative survival probability on the Y-axis indicates the cumulative delinquency maintenance probability, which came close to 40% in the first month, but dropped to almost 20% in the second month.

4.2 Evaluation of the Cox Model

This study was interested in correctly predicting when credit delinquents would escape from the delinquency state. Therefore, models deploying C5.0 and neural net (NN) techniques were developed for making comparisons with the Cox model. The same data set that had been applied to the Cox model was also used for each comparative model. First, C5.0 adopted a boosting/cross-validation method and a local/global pruning in training. A branch of the tree was split only if the sub-branches contained at least five records. The actual credit recovery month was the target response variable and entropy was the impurity measurement. NN placed five neurons in the input layer, 20 neurons in the hidden layer, and 11 neurons in the output layer. NN set the learning rate to 0.9 and the momentum to 0.3, which were decaying gradually to 0.01.

The models were compared in terms of their gains. Fig. 2 shows the gain charts for the three models being compared. The gain charts effectively illustrate, for each decile, the percentage of predicted events (i.e., credit recovery). The diagonal line plots the expected response for the entire sample if the models are not used (in this case, the response rate would be constant). The curved lines indicate how much the models can improve the response based on gain functions. The larger the area between the gain chart and the diagonal line, the better the model under consideration. From Fig. 2, it appears that the gain chart for the Cox model is better than any other models, and it shows the best performance.

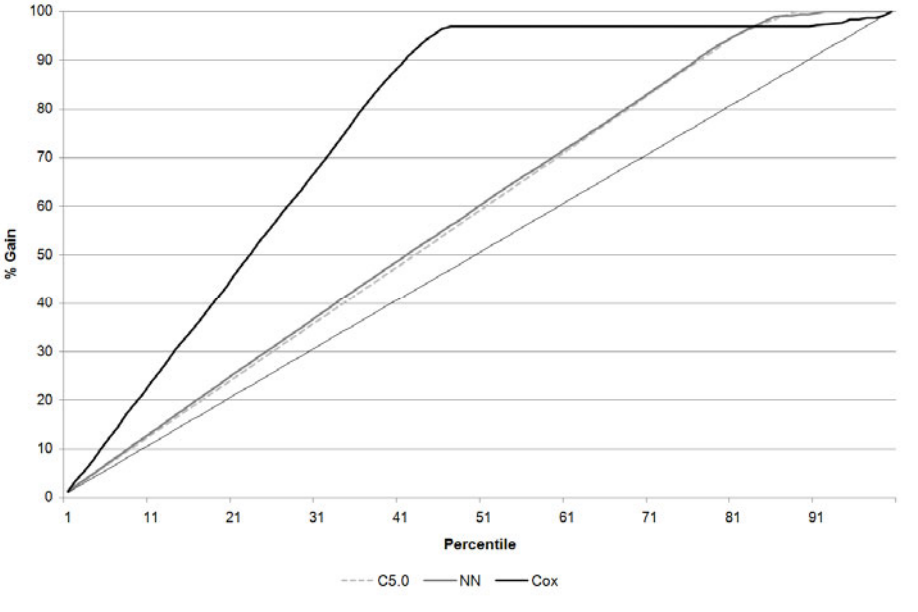


Fig. 2. Gain charts for the models considered, based on the validation data

Furthermore, this study calculated the prediction accuracy rates for the models: all rates were calculated after performing 20-time experiments using different random partitioning of data. On the training set, the Cox model has the highest accuracy, followed by the NN model, and then the C5.0 model, as shown in Table 2. On the validation set, the Cox model also has the highest accuracy, followed by the NN model, and then the C5.0 model. To summarize, the Cox seems to be the best-performing model.

Table 2. Prediction accuracy rates for each model

		Cox	C5.0	NN
Training data	Mean	93.34%	89.06%	89.29%
	Variance	5.8E-06	1.1E-05	2.1E-06
Validation data	Mean	93.24%	88.57%	90.23%
	Variance	0.003	0.005	0.000

Table 3 shows the confusion matrix for the Cox model obtained on the validation data set. The Cox model predicts as delinquent customers that are, in fact, non-delinquent in 1.28% of the cases; the type I error. On the other hand, it predicts as credit-win-back customers that are actually delinquent in 5.48% of the cases; the type II error. Notice that the total misclassification rate is about 6.76%.

Table 3. Prediction confusion matrix for the Cox model

		Prediction	
		Delinquency	Credit recovery
Observation	Delinquency	0.5165	0.0548
	Credit recovery	0.0128	0.4159

5 Conclusion

This study presented a credit classification method to solve the common problems of credit delinquency in a typical retail store. The credit classification method used a Cox regression model to analyze the patterns of credit recovery. The number of delinquency and the number of repayments were the most predictive variables. As the number of delinquency increases, the amount of time required for credit recovery increases. The amount of time for credit recovery from delinquency decreases as the number of repayments for credit recovery increases. Using these variables, the Cox regression model showed a predictive power of 93.24%.

The contributions of this study are summarized as follows. First, while most previous research on credit prediction has focused on classifying customers into two groups of good/bad credit customers, this study made it possible to analyze the delinquent customers who had recovered from a credit delinquency state to a good credit state. Second, this study used a survival analysis to identify the influential variables on the rate of credit recovery and to predict the time of credit recovery.

This study, however, has some limitations that will be improved in future research. First, it used only 12 months of data associated with the credit delinquents. This made it difficult to generalize the research results. Therefore, various types of variables for multiple periods need to be collected and analyzed for generalization of this type of research. Second, this study focused on those delinquents who had recovered from a credit delinquency state. However, a future study needs to encompass all kinds of customers, from good to bad credit, to develop a more complete credit prediction system.

References

1. Chen, M.C., Huang, S.H.: Credit scoring and rejected instances reassigning through evolutionary computation techniques. Expert Syst. Appl. 24, 433–441 (2003)
2. Crook, J.N., Edelman, D.B., Thomas, L.C.: Recent developments in consumer credit risk assessment. Eur. J. Oper. Res. 183, 1447–1465 (2007)
3. Bradley, P.S., Fayyad, U.M., Mangasarian, O.L.: Mathematical programming for data mining: formulations and challenges. INFORMS J. Comput. 11, 217–238 (1999)
4. Min, J.H., Lee, Y.-C.: A practical approach to credit scoring. Expert Syst. Appl. 35, 1762–1770 (2008)
5. Tsaih, R., Liu, Y.-J., Liu, W., Lien, Y.-L.: Credit scoring system for small business loans. Decis. Support Syst. 38, 91–99 (2004)
6. Allen, L.N., Rose, L.C.: Financial survival analysis of defaulted debtors. J. Oper. Res. Soc. 57, 630–636 (2006)
7. Sohn, S.Y., Shin, H.W.: Reject inference in credit operations based on survival analysis. Expert Syst. Appl. 31, 26–29 (2006)
8. Kumar, P.R., Ravi, V.: Bankruptcy prediction in banks and firms via statistical and intelligent techniques-a review. Eur. J. Oper. Res. 180, 1–28 (2007)
9. Hu, X.: A data mining approach for retailing bank customer attrition analysis. Appl. Intell. 22, 47–60 (2005)
10. Chen, M.-C., Chen, L.-S., Hsu, C.-C., Zeng, W.-R.: An information granulation based data mining approach for classifying imbalanced data. Inform. Sciences. 178, 3214–3227 (2008)
11. Huang, J.-J., Tzeng, G.-H., Ong, C.-S.: Two-stage genetic programming (2SGP) for the credit scoring model. Appl. Math. Comput. 174, 1039–1053 (2006)
12. Abdou, H., Pointon, J., El-Masry, A.: Neural nets versus conventional techniques in credit scoring in Egyptian banking. Expert Syst. Appl. 35, 1275–1292 (2008)
13. West, D.: Neural network credit scoring models. Comput. Oper. Res. 27, 1131–1152 (2000)
14. Malhotra, R., Malhotra, D.K.: Evaluating consumer loans using neural networks. Omega-Int. J. Manage. S. 31, 83–96 (2003)
15. Hsieh, N.-C.: An integrated data mining and behavioral scoring model for analyzing bank customers. Expert Syst. Appl. 27, 623–633 (2004)
16. West, D., Dellana, S., Qian, J.: Neural network ensemble strategies for financial decision applications. Comput. Oper. Res. 32, 2543–2559 (2005)
17. Yen, E.C.-C.: Warning signals for potential accounting frauds in blue chip companies – An application of adaptive resonance theory. Inform. Sciences 177, 4515–4525 (2007)
18. Thomas, L.C.: A survey of credit and behavioral scoring: Forecasting financial risk of lending to consumers. Int. J. Forecasting 16, 149–172 (2000)

19. Hansen, J.V.: Combining predictors: comparison of five meta machine learning methods. Inform. Sciences 119, 91–105 (1999)
20. Hsieh, N.-C.: Hybrid mining approach in the design of credit scoring models. Expert Syst. Appl. 28, 655–665 (2005)
21. Chen, W., Ma, C., Ma, L.: Mining the customer credit using hybrid support vector machine technique. Expert Syst. Appl. 36, 7611–7616 (2009)
22. Martens, D., Baesens, B., Van Gestel, T., Vanthienen, J.: Comprehensible credit scoring models using rule extraction from support vector machines. Eur. J. Oper. Res. 183, 1466–1476 (2007)
23. Lee, T.S., Chiu, C.C., Lu, C.J., Chen, I.F.: Credit scoring using the hybrid neural discriminant technique. Expert Syst. Appl. 23, 245–254 (2002)
24. Lee, T.S., Chiu, C.C., Chou, Y.C., Lu, C.J.: Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. Comput. Stat. Data An. 50, 1113–1130 (2006)
25. Zhu, Z., He, H., Starzyk, J.A., Tseng, C.: Self-organizing learning array and its application to economic and financial problems. Inform. Sciences 177, 1180–1192 (2007)
26. Laha, A.: Building contextual classifiers by integrating fuzzy rule based classification technique and k-nn method for credit scoring. Adv. Eng. Inform. 21, 281–291 (2007)
27. Greene, W.H.: Econometric Analysis, 6th edn. Prentice-Hall, Upper Saddle River (2007)

An Ontology-Based Adaptive Learning System to Enhance Self-directed Learning

Mihye Kim[1] and Sook-Young Choi[2,*]

[1] Department of Computer Science Education, Catholic University of Daegu,
330 Hayangeup Gyeonsansi Gyeongbuk, South Korea
mihyekim@cu.ac.kr
[2] Department of Computer Education, Woosuk University,
490 Samrae-eup, Jeonbuk, South Korea
sychoi@woosuk.ac.kr

Abstract. An adaptive learning system provides learning content customized to a student's needs and characteristics, such as in terms of knowledge level, preferences, and learning style. Individualization is essential for improved learning experiences. Here we propose such an adaptive learning system based on ontologies that can assemble highly personalized learning content from implied inter-ontology relationships, resulting in a seamless linking to a more enhanced self-directed learning environment. A prototype was developed for a biology content domain used by second-grade students in Korea. The system presented learning content through a visualized hierarchical interface to enhance student understanding.

Keywords: Adaptive learning, ontology-based learning.

1 Introduction

Rapid advances in information and communication technology (ICT) are changing the way people access information in an environment where much information is easily accessible. With such transformation of the knowledge-based environment, e-learning has become a main teaching and learning method, and the current educational environment is shifting into a new pedagogical paradigm with various forms of web-based educational content.

To successfully employ e-learning, a web-based education system should choose and organize useful information and provide it in a searchable format. It should also support a self-directed learning environment that can facilitate students' self-regulated learning without direct supervision or guidance from teachers or fellow students [1]. In addition, it should address the problems of cognitive overload and disorientation of students by providing adaptive presentation and navigation that take into consideration students' background knowledge, preferences, and learning styles [2], [3]. An adaptive learning system can reduce students' unnecessary searching and external

* Corresponding author.

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 91–102, 2010.

cognitive load [4]. The most commonly used techniques for implementing an adaptive learning system are adaptive hypermedia [5], concept maps [6], and ontologies in the Semantic Web environment [7–13].

An ontology-based adaptive learning system is based on Semantic Web technology. The Semantic Web, known as the next-generation intelligent web, aims to provide more accurate information and enhanced intelligent web services by encoding machine processable semantics in the Web content. Here, ontologies play the core role of encoding semantics for Web resources. In computer science, an ontology is defined as 'a formal, explicit specification of a shared conceptualization' for a specific domain [14], and is used as a mechanism for knowledge representation. Typically, a formal ontology consists of concepts, definitions, and formal axioms relating them together. Those definitions associate the components of entities, such as classes (including their subclasses and attributes), relations, functions, and constraints [14]. Ontologies not only enable to support more accurate searches, but also empower more advanced information services by deriving new concepts automatically from inter-ontology relationships.

Accordingly, based on the conceptual relationships among them, learning content can be organized into a knowledge structure using ontology technology [15]. The explicit representation of learning content can enable a higher level of learning service by establishing an intelligent query that infers implicated knowledge in ontologies [16]. Furthermore, it is useful to provide personalized learning content to infer learning elements appropriate to each learner's characteristics from both a domain ontology and user model ontology [1].

The aim of this study is to design and propose an ontology-based adaptive learning system that assembles highly personalized learning content based on implied inter-ontology relationships. The system consists of three main ontologies: domain, content structure, and user model ontologies. To provide a more precise, individualized learning service appropriate to the characteristics of each student, it uses a user diagnostic module that deduces and analyzes students' characteristics such as level of understanding, learning styles, preferences, learning history, and learning context. Then, the system dynamically builds personalized learning content for a particular student based on the inferred characteristics.

The remainder of this paper is organized as follows. Section 2 reviews the existing literature on adaptive learning systems. Section 3 presents the proposed ontology-based adaptive learning system. In Section 4, a prototype for middle school biology is presented as an application of the proposed system. The paper concludes with a discussion on possible directions for future research.

2 Related Work

An adaptive learning system provides learning content that is customized to each student's level of knowledge, preferences, and learning style, experience, and history. It offers learning content mainly using an adaptive hypermedia system or by hiding links to content according to a student's characteristics [17], [18]. The most commonly used techniques for implementing an adaptive learning system are adaptive hypermedia [5], concept maps [6], and/or ontologies in the Semantic Web

environment [7–13]. But some ontology-based systems [7], [9] present learning content using adaptive hypermedia and/or concept maps based on domain ontologies.

To provide learning content more adaptively, not only content but also learner information should be managed in a more semantic and systematic way. Under a well-organized knowledge structure, more meaningful and individualized learning content can be inferred and serviced to learners. As a result, recent studies have actively focused on ontology-based adaptive learning systems.

Some adaptive learning systems based on ontologies are AIMS [7], [8], LAOS [9], [10], OntAware [11], Personal Reader [12], and TANGRAM [13]. AIMS (agent-based information management system) [7], [8] supports adaptive learning using concept maps, which conceptualize learning processes based on domain ontology. It provides learning content and navigation adapted to each student's profile and level of knowledge. It assembles personalized navigation content from the domain ontology according to the interrelations between the concepts that the student has already studied and wishes to study in the ontology.

LAOS (layered adaptive hypermedia system [AHS] authoring-model and operators) [9], [10] is a layered model for adaptive hypermedia authoring based on concept maps and AHS. This model consists of five layers: the domain (DM), goal and constraints (GM), user (UM), adaptation (AM), and presentation (PM) models. One domain map can be adapted differently according to various pedagogic goals, and learning content can be presented differently based on one or more adaptive strategies corresponding to the pedagogic strategy.

OntoAWare provides adaptive e-Learning content authoring, management, and delivery based on Semantic Web technologies [11]. It enables semi-automatic generation of learning objects from domain ontologies and offers ontology-based navigation to students in real time. It determines the personalized navigation path for each student based on two criteria: the student's level of knowledge determined by a test given before and after learning, and the concepts that the student has clicked on for learning (i.e., after visiting a particular concept, the system assumes that the student knows the concept and does not need to revisit it). Based on these two assumptions, the system provides adaptive guidance to the student.

Personal Readers [12] were developed to support individualized e-learning in the Semantic Web. They provide personalized contextual information to users on the current learning object, such as additional reading recommendations, more detailed information, quizzes, and exercises. They are based on four ontologies: a domain ontology that defines the elements (classes) of learning content and the conceptual relationships among the elements, a user model ontology that provides the user's characteristics and user device information, an observation ontology that defines various interaction information of user observation detected in real time, and an ontology for describing adaptive functionality for adaptive services. Based on these ontologies, the system provides learning content adapted to the user's characteristics.

TANGRAM [13] also provides personalized learning content based on Semantic Web technology. It generates new learning content customized to the specific needs of an individual student, and assembles new content adapted to the student's current level of domain knowledge, preferences, and learning style. TANGRAM defines three ontologies for adaptive learning: a domain ontology for representing learning object content structure (i.e., to define semantic relationships between topics of a course in a

hierarchy), a learning paths ontology to specify pedagogical relations among domain concepts, and a user model ontology to represent user information. The student is required to complete a questionnaire and exam that determine learning style and knowledge level. Based on these deduced results and the ontologies, the system builds a visual representation of learning content adapted to the user in the form of a tree of links, exploiting link annotation and link-hiding techniques.

Most adaptive learning systems based on ontologies provide a domain ontology and a user ontology for adaptive support. They also provide learning content adapted to students based on their prerequisites and learning histories, as well as their characteristics such as levels of knowledge, preferences, and learning styles. The current approaches, however, do not provide any effective diagnostic modules that can more accurately and synthetically analyze students' learning circumstance and characteristics while their learning proceeds. The learning systems consider the main properties of adaptive learning such as students' level of knowledge, learning styles, learning context, but they do so only partially. Moreover, while TANGRAM and LAOS consider each student's learning style based on user ontology, in reality, they do not support adaptive learning according to learning style in a more specific and systematic way through the overall learning process. Therefore, this study proposes an adaptive learning system that addresses such issues by taking into account the advantages of ontology-based approaches. This work expanded earlier work [1], [19].

3 An Ontology-Based Adaptive Learning System

Individualized learning systems adaptively provide learning content according to each student's level of knowledge, preferences, and other considerations. Individualization is essential for improved learning experiences. Here we propose an adaptive learning system that can provide highly individualized learning content based on ontologies. The architecture of the proposed ontology-based adaptive learning system consists of three main ontologies and four modules as shown in Fig. 1.

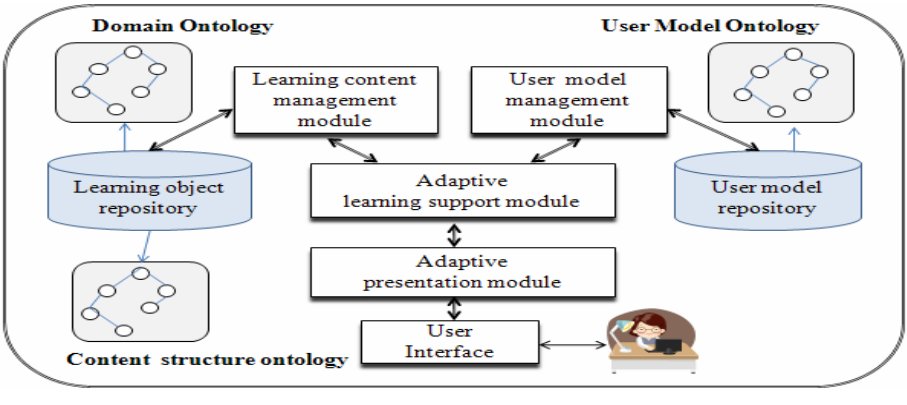


Fig. 1. Architecture of the proposed ontology-based adaptive learning system

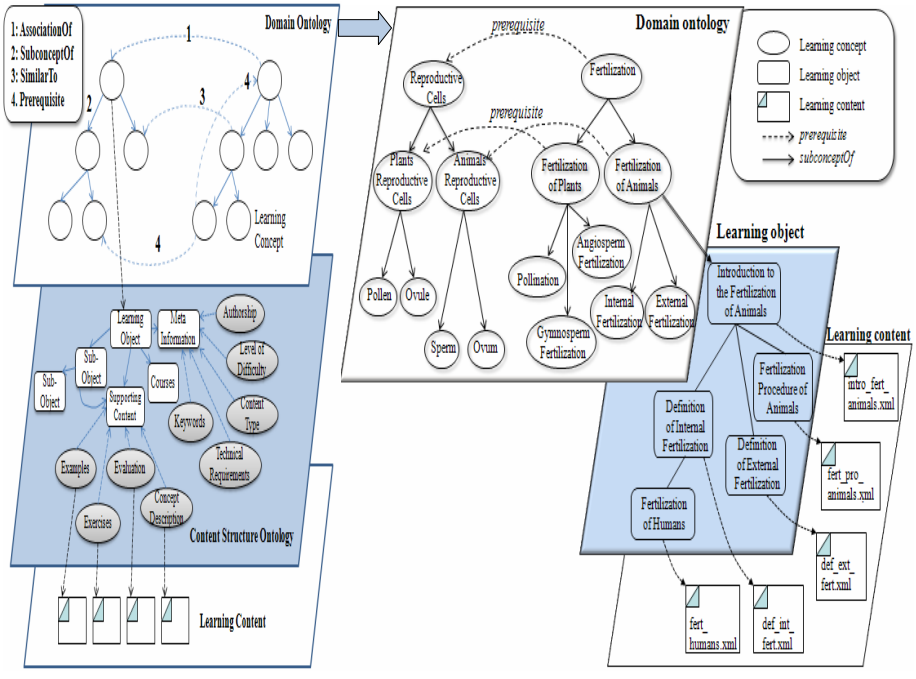


Fig. 2. Hierarchical tree structure of the domain and content structure ontologies for a biology domain

The domain and content structure ontologies are constructed as hierarchical trees and relate directly to the learning content in the evolved domain as shown in the left-hand diagram of Fig. 2. The right-hand diagram in Fig. 2 shows an example structure developed for a biology domain used in Korean middle school classes. The user model ontology represents student information (Fig. 3), and the four modules are coordinated with three ontologies for adaptive learning in a manner similar to that used in the TANGRAM model [13].

Domain Ontology. The domain ontology defines learning concepts for a specific domain and establishes relationships between the concepts. We use four relational properties to specify relations between learning concepts: AssociationOf, SubconceptOf, SimilarTo, and Prerequisite. The *AssociationOf* property defines association relationships, and *SubconceptOf* defines sub-concept relationships between concepts. *SimilarTo* specifies similar relationships, and *Prerequisite* defines prerequisite relationships between concepts. Through these relationships, the system can more easily deduce and assemble (in real time) learning content based on the highly subjective needs and current learning status of each student. For example, in the right-hand diagram of Fig. 2, the concept ‘Reproduction Cells’ is a prerequisite concept of the concept ‘Fertilization’. And, the concepts ‘Fertilization of Plants’ and ‘Fertilization of Animals’ are sub-concepts of the concept ‘Fertilization’.

Content Structure Ontology. The content structure ontology represents the knowledge structure of learning objects for each learning concept. It provides information on actual learning objects that correspond to each learning concept and is used to support adaptive learning appropriate to each student's needs and characteristics.

User Model Ontology. The user model ontology represents the characteristics of each student. It includes each student's learning history, learning level, prerequisite knowledge, learning goals, learning time, learning context, and learning style as shown in Fig. 3. Learning context includes the operating system, terminal type, and network information for the student's computer. We analyzed a student's learning style based on the four dimensions suggested by Felder-Silverman [20].

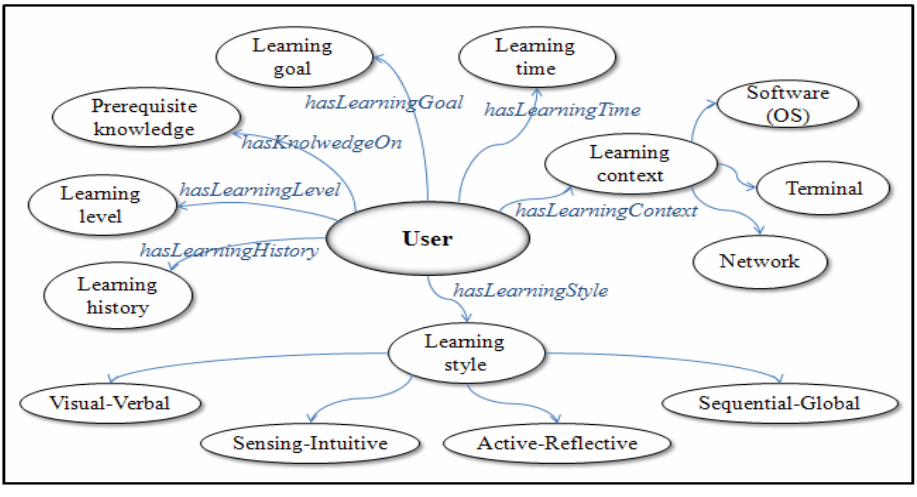


Fig. 3. User model ontology

Learning Content Management Module. The learning content management module manages the repository of learning objects. It inserts a new learning object into the repository or manipulates existing learning objects. Each constructed learning object corresponds to each learning concept in the hierarchy of the content structure ontology as shown in Fig. 2. A parent object generally consists of a number of sub-child concepts.

User Model Management Module. This module manages the repository of user models, responding to user information requests such as inserting into, updating, or accessing the repository.

Adaptive Learning Support Module. This module dynamically generates personalized learning content for a particular student. In association with the user model management module, it diagnoses the knowledge level of a student based on a questionnaire and student test results. It also analyzes the student's learning history, learning context, and learning style. Based on these acquired student characteristics, it

guides the student learning process using the most appropriate learning content. We have also defined some inference rules to support adaptive learning.

Adaptive Presentation Module. The adaptive presentation module is responsible for presenting individualized learning content to a specific student based on results from the adaptive learning support module. The system presents adaptive learning content using link-hiding techniques. If analyses flag certain content as inappropriate for a student, the link to that learning content is hidden from the student. For example, if a student has not learned about or does not have a good understanding of the concept 'Fertilization of Animals,' the links to 'Internal Fertilization' and 'External Fertilization' are hidden so that the student cannot select those concepts in Fig. 2.

4 A Biology Domain Prototype

We have developed a biology domain prototype for second-grade students in Korea as a test application of the proposed protocols. The system was designed to implement an enhanced self-directed learning environment by providing learning content directly related to each student's characteristics. Consequently, the students themselves could directly control their learning process.

4.1 Construction of Ontologies

The ontologies (domain, content structure, user model) were represented using the Web Ontology Language (OWL) and constructed using the Protégé-OWL editor, an extension of Protégé that supports OWL. For inference rule representation, we used the Semantic Web Rule Language (SWRL) and Jess as an inference engine. Jess is Java-based rule engine written by Ernest Friedman-Hill [21] and is characterized as being small, light, and one of the fastest rule engines available [22].

Figures 4 and 5 show domain ontology example constructs and their instances (i.e., learning objects), respectively, in the Protégé editor. The topic was 'Reproduction Cells and Fertilization' as used in middle school classes.

4.2 Process of Adaptive Learning

During initial registration process, the student is required to fill out a questionnaire for the initial setting of his/her user model, especially to determine his/her learning style. We used the questionnaire developed by Felder-Silverman to infer a student's learning style [20], [23]. Learning styles are divided into 16 categories by using exclusive combinations of the eight types with each of the four dimensions suggested by Felder-Silverman (i.e., active/reflective, sensing/intuitive, visual/verbal, sequential/global). Depending on students' learning styles, the system provides customized learning forms in such respects as learning methods and construction, internal representation, and presentation of learning content. At this stage, the student is required to conduct a self-assessment to judge the knowledge level of learning concepts that they are subjected. The student's level is represented with a value using an overlay model.

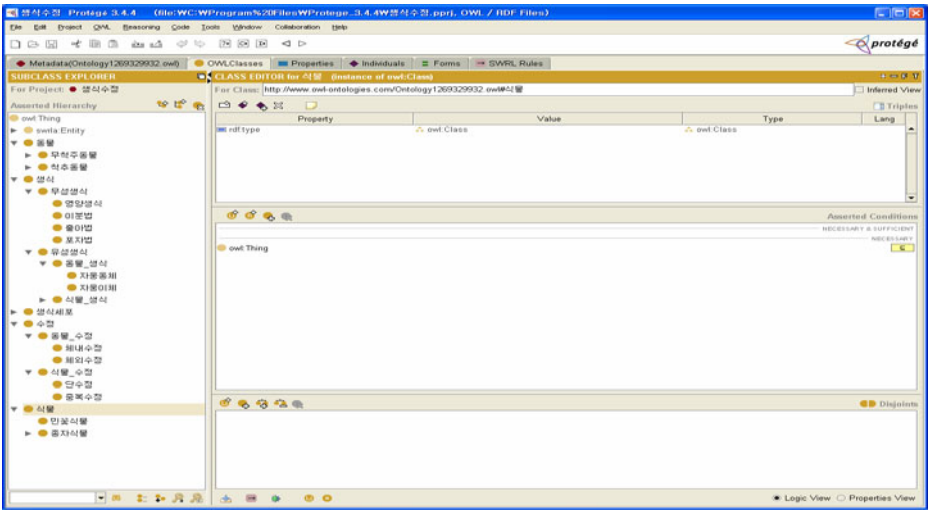


Fig. 4. Example of defining a domain ontology using the Protégé editor

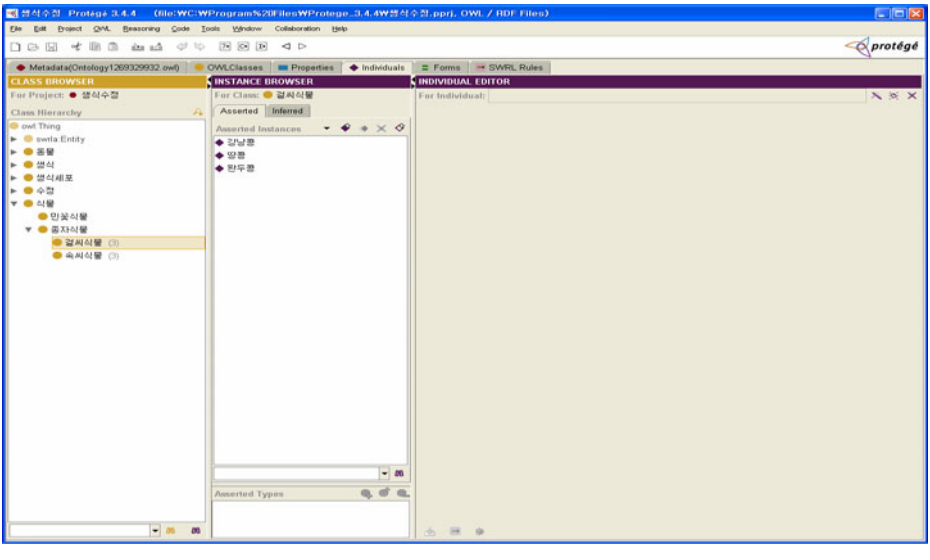


Fig. 5. Example of defining the instances of the domain ontology

Next, when the student selects a learning topic during the initial access to the system, once again, the student is required to perform a self-assessment to determine the level of his or her knowledge of the topic. The self-assessment process offers three response options: 'I do not know about it at all,' 'I have a basic knowledge of it,' and 'I know it well.' At this stage, the system also analyzes information about user learning context, such as operating system, terminal type, and the network information

of the student's computer. Then, the system dynamically builds individual learning content for the student by taking into account the student's learning styles, level of knowledge and learning context derived. The system also updates the student's user model accordingly.

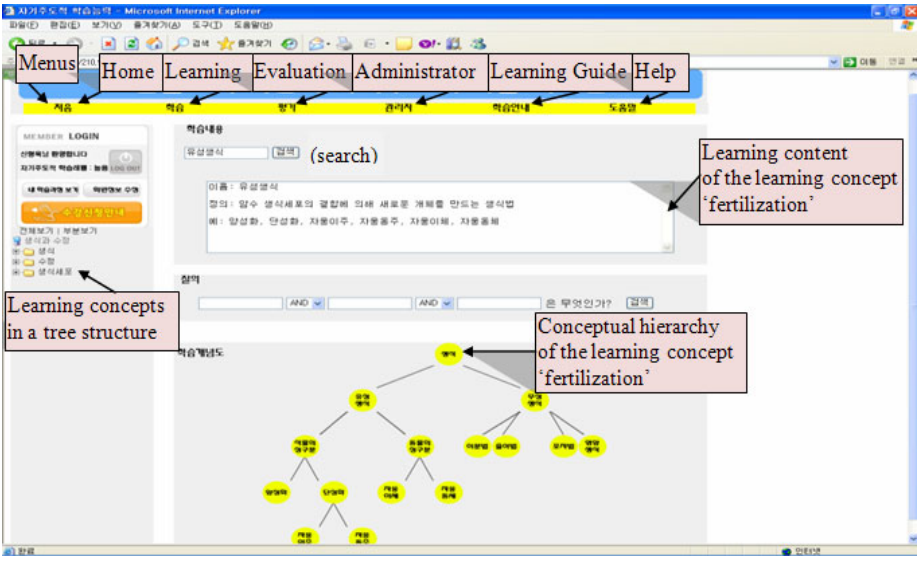


Fig. 6. Example of the self-directed learning system for the concept 'fertilization'

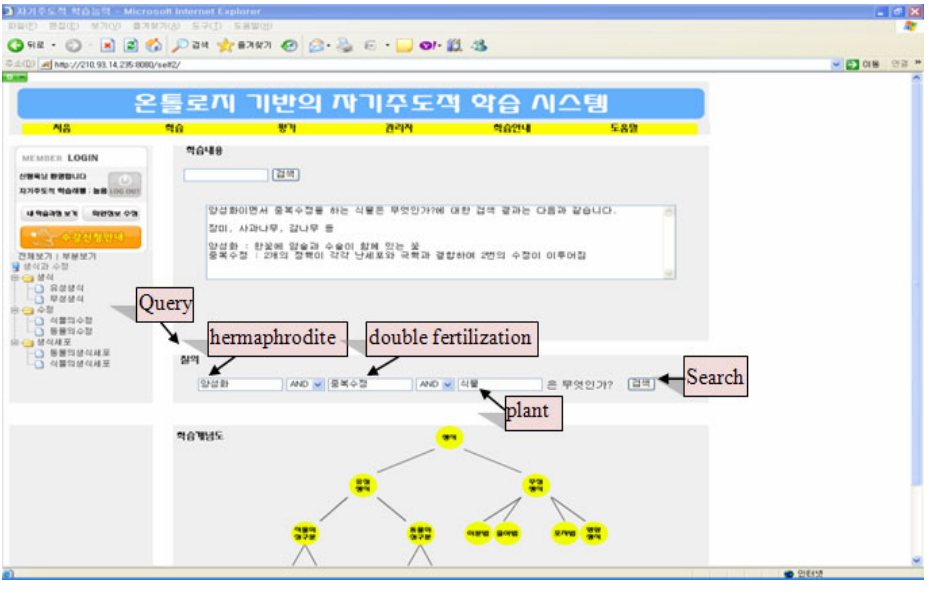


Fig. 7. Example of a query to obtain more specific learning content

With each subsequent accessing of a topic, the system refers to the user model ontology to obtain the student's characteristics and level of knowledge about the selected topic. The system also analyzes the prerequisite concepts of the selected topic through the prerequisite property of the domain ontology and determines the student's learning condition. If the student has not yet learned the topic or does not have sufficient knowledge of prerequisite concepts, the system hides links to the topic and guided the student through prerequisite concepts first. If the student has enough knowledge of prerequisite concepts and has learned the basics of the topic, then the system presents higher level content. After the learning process, the system performs an evaluation and updates the relevant ontologies accordingly.

Fig. 6 and Fig. 7 show examples of the self-directed learning system based on the proposed approach. Fig. 6 shows an example for learning on the concept 'fertilization'. The system presents sub-concepts of the current learning concept through a graphical interface. The student can easily explore the current concept through the visual hierarchy. Visual representation not only can enhance a student's understanding of learning content but also can allow the student to learn in a more effective way. The left-side of Fig. 6 shows the learning concepts presented in a traditional hierarchical tree structure.

The student can specify a query by entering search words related to the learning content using a conventional information retrieval process. The system infers and provides search results from the interrelationships of the domain and user model ontologies. This offers ontology-based navigation to students. When a student wants to learn a more specific or more advanced concept of the domain, he or she can formulate a query using any text words with the AND Boolean operator. Fig. 7 shows the query result with the terms 'hermaphrodite', 'double fertilization', and 'plant'; that is, it is the search result of the question: "Which plants are hermaphrodite with double fertilization?" Results are derived from the interrelations among the learning concepts in the ontologies and inference rules defined for adaptive learning. Then, the system builds a visual representation of the new learning content and presents it to the user. This learning environment can support more effective e-learning and enhance the student's interest and motivation to learn, resulting in the seamless linking to a more enhanced self-directed learning environment.

5 Conclusion

We have proposed an ontology-based adaptive learning system and developed a prototype biological content domain for use in Korean middle school classes. The objective was to develop an adaptive learning system that provided highly personalized learning content based on conceptual relationships within the domain ontology. In addition, the system analyzed students' characteristics in a semantic and synthetic way throughout the learning process. Each student's level of knowledge, learning history, knowledge of prerequisites, learning goals, learning style, and learning context were included in the user model ontology. The system consisted of three main ontologies: domain, user model, and content structure.

The explicit representation of learning content within a knowledge structure of ontology can engender a higher level of learning by establishing intelligent student

queries. In addition, through ontological relationships the system can more easily and accurately deduce (in real time) learning content pertinent to the current learning status of a student. Moreover, it can support a more enhanced self-directed learning environment such that students can freely and successfully control the learning process. The system proposed here presents adaptive learning content as a hierarchical tree of links based on the concept structure of the domain ontology. We believe that this visual representation of learning content not only can enhance a student's understanding but also can allow learning to progress in a more effective manner. A student can formulate a key word query to access more specific learning content or higher level knowledge during the learning process.

Although this ontology-based application may be useful in preparing the way for adaptive learning based on ontology technology, a number of issues still remain. These issues relate to the further development of the system and also to further evaluation of the system through practical classroom trials. Also, there should be ongoing investigations into improvements suggested by students via a wider range of case studies. We have not yet fully developed and evaluated the proposed approach. However, we believe that our results are valuable because this approach can lead students into a more enhanced self-regulated learning environment by taking into account ontological advantages and by more accurately and synthetically analyzing students' learning characteristics.

References

1. Choi, S.Y.: Application of Ontology technology for Adaptive Learning in e-Learning. Journal of Korean Association of Computer Education 12(6), 53–67 (2009)
2. Bra, P.D., Brusilovsky, P., Houben, G.J.: Adaptive Hypermedia: From Systems to Framework. In: ACM Computing Survey, Symposium Edition, vol. 41(4es) (December 1999)
3. Wu, H., Kort, E.D., Pra, P.D.: Design Issues for General-Purpose Adaptive Hypermedia Systems. In: Proceedings of the ACM Conference on Hypertext and Hypermedia, Aarhus, Denmark, pp. 141–150 (August 2001)
4. Jung, H.S., Park, S.B.: A Web-based adaptive hypermedia system for novices to learn programming. Journal of Korean Association of Computer Education 7(6), 37–45 (2004)
5. Brusilovsky, P., Millan, E.: User Models for Adaptive Hypermedia and Adaptive Educational Systems. The Adaptive Web, 3–53 (2007)
6. Choi, S.Y.: An Adaptive Tutoring System using Concept-Map. Journal of Korean Association of Computer Education 9(1), 29–39 (2006)
7. Aroyo, L., Dicheva, D.: AIMS: Learning and Teaching Support for WWW-based Education. Int. J. for Continuous Eng. Education and Life-long Learning 11(1/2), 152–164 (2001)
8. Aroyo, L., Dicheva, D., Cristea, A.: Ontological support for Web Courseware Authoring. In: Cerri, S.A., Gouardéres, G., Paraguaçu, F. (eds.) ITS 2002. LNCS, vol. 2363, pp. 270–280. Springer, Heidelberg (2002)
9. Cristea, A., De Mooij, A.: LAOS: Layered WWW AHS Authoring Model and its corresponding Algebraic Operators. In: Proceedings of WWW 2003, Budapest, Hungary. ACM, New York (2003)
10. Cristea, A.: What can the Semantic Web do for Adaptive Educational Hypermedia? Educational Technology & Society 7(4), 40–58 (2004)

11. Holohan, E., Melia, M., McMullen, D., Pahl, C.: Adaptive E-Learning Content Generation based on Semantic Web Technology. In: SW-EL 2005: Application of Semantic Web Technology for E-Learning (in conjunction with AIED 2005), pp. 29–36 (2005)
12. Henze, N.: Personal Readers: Personalized Learning Object Readers for the Semantic Web. In: Proceeding of the 12th International Conference on Artificial Intelligence in Education (AIED 2005), pp. 15–22 (2005)
13. Jovanovic, J., Gasevic, D., Devedzic, V.: TANGRAM for Personalized Learning Using the Semantic Web Technologies. Journal of Emerging Technologies in Web Intelligence 1(1), 6–21 (2009)
14. Gruber, T.R.: Adaptive A Translation Approach to Portable Ontology Specification. Knowledge Acquisition 5(2), 199–220 (1993)
15. Mcguinness, D.L., Van Harmelem, F.: OWL web Ontology Language Overview (2009), http://www.w3.org/TR/owl-features/
16. Koper, R.: Use of the Semantic Web to Solve Some Basic Problems in Education: Increase Flexible, Distributed Lifelong Learning, Decrease Teacher's Workload. Journal of Interactive Media in Education, Special Issue on the Educational Semantic Web 2004(6) (2004)
17. Wolf, C.: iWeaver: towards 'learning style'-based e-learning in computer science education. In: Proceedings of the fifth Australasian Conference on Computing Education (ACE 2003), Adelaide, Australia, pp. 273–279 (2003)
18. Thyagharajan, K., Nayak, R.: Adaptive Content Creation for Personalized e-Learning Using Web Services. Journal of Applied Sciences Research 3(9), 828–836 (2007)
19. Choi, S.Y.: An Ontology-based Learning System for Supporting Self-directed Learning in Online Environments. In: Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2009, pp. 2459–2464 (2009)
20. Felder, R., Silverman, L.: Learning and Teaching Styles in Engineering Education. Journal of Engineering Education 78(7), 674–681 (1988)
21. Friedman-Hill, E.: Jess in Action. Manning Publications Co., Greenwich (2003)
22. Jess, the Rule Engine for the Java Platform (2008), http://www.jessrules.com/jess/index.shtml
23. Soloman, B.A., Felder, R.M.: Index of Learning Style Questionnaire. North Carolina State University (2003), http://www.engr.ncsu.edu/learningstyles/ilsweb.html

Context-Aware Service Framework for Decision-Support Applications Using Ontology-Based Modeling

Giovanni Cagalaban and Seoksoo Kim*

Department of Multimedia, Hannam University, 306-791 Daejeon, Korea
gcagalaban@yahoo.com, sskim0123@naver.com

Abstract. The technological advances in healthcare specifically in preventive healthcare can lead to longer life expectancy especially for the elders. To aid in preventing premature loss of lives as well as lengthening life span, this research aims to implement the use of mobile and sensor technology to improve the quality of life and lengthen life expectancy. In particular, we propose a context-aware service framework for decision-support applications in preventive healthcare. This research applies ontology to support context modeling and reasoning by applying the general application areas for ontologies to the domain of context in ubiquitous computing environments. This paper also demonstrates how context technologies and mobile web services can help enhance the quality of services in preventive healthcare to elders.

Keywords: context-aware, decision-support, context modeling, context reasoning.

1 Introduction

Context awareness is vital for ubiquitous computing environments to adapt computational entities to changing situations. It has been drawing much attention from researchers since it was proposed about a decade ago. However, context-aware services have never been widely available to everyday users. The concept of context-awareness refers to knowledge and understanding of the surrounding environment within which the decision support system has to operate.

While notion of context is interpreted by computer science researchers somewhat differently, we subscribe here to the definition of context as adopted by the ubiquitous computing research community. That is, context is defined as 'the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user' [1]. Within the context of real-time decision support, it is imperative to equip the decision maker with full information based on best possible context model [2]. For instance, the awareness of a person in making a decision or the approximate time to get a patient to a hospital where he/she will get best care, are context attributes.

This paper aims to propose a context aware service framework for decision-support applications for the highly dynamic and changing healthcare domain especially for

* Corresponding author.

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 103–110, 2010.

preventive healthcare of elders. In this field, various researches are done towards finding ways to prevent an illness or injury to happen. Specifically, this research aims to demonstrate the use of mobile-based agent to support the deployment of a decision-support system in real-time situations. This research focuses on integrating context awareness, mobile software agents and decision support concepts. We solve the policy problem by classifying context using context modeling and implementing context reasoning. In offering adaptive services, we selected mobile-based agents as the key enabling technology because they offer a single, general framework in which large scale distributed real-time decision support applications can be implemented more efficiently.

The rest of this article is organized as follows. Section 2 discusses related works. In section 3, we present the application scenario of the context-aware system. Section 4 follows with the concepts of context modeling and reasoning that we implemented. It is followed by the system architecture in Section 5. Section 6 presents the evaluation of the system and lastly, section 7 presents the conclusion of the article.

2 Related Works

The concept of context has been categorized in several ways by different researchers in this area. In general, context refers to information about the device platform, the user, and the surrounding environment. Several context-aware systems have been developed to demonstrate the usefulness of context-aware computing technology. Early works focused on building application-specific context-aware systems such as the Cyberguide [3] project which provided a context-aware tour guide to visitors. The ContextToolkit used an object-oriented approach to provide a framework and a number of reusable components to support rapid prototyping of sensor-based context-aware applications. But these systems do not provide a common context model to enable context knowledge sharing and context reasoning.

Recently, research works have focused on providing infrastructure support for context-aware systems. The advantage of the infrastructure-based systems has been pointed out in [4]. A middleware for context awareness and semantic interoperability was developed by [5] in which they represented context ontology written in DAMLC + OIL [6]. Context has been used in distributed event-based applications for wide-area networks. These applications are characterized by a publish/subscribe mode [7], [8].

3 Application Scenario

In this research, we first illustrate the application scenario in the field of preventive care for the context-aware service framework. Consider Mr. Kim, an elder who has a systemic, arterial hypertension and needs to check his blood pressure regularly. One solution is to keep his blood pressure under control. To do that, he follows a strict diet prescribed by his nutritionist. However, as most elders do, they have a hard time keeping up with their health regime. He also needs to do routine medication according to his health conditions but sometimes he tends to forget to take in medicines on time.

In some cases, he wants to eat good but unhealthy foods. As a result, there is a need to monitor his dietary behavior as well as his physical activities.

Such cases can be improved with the aid of intelligent environment technology which includes sensors, context-aware reasoning and web services where it can provide personal reminders for health measurements, meals and medication. It can also issue health alerts to that timely medical assistance will be provided during emergency cases. Mobile devices can be used to monitor the elder's general activities, meals and medication anytime and anywhere.

4 Context Modeling and Reasoning

This section describes the concept of ontology and the OWL Web Ontology Language. It also describes the ontology-based approach for modeling contexts and the approach to further deduce context information through context reasoning.

4.1 Ontology Using OWL

Ontology is a formal explicit description of concepts and provide for representing knowledge about a domain and for describing specific situations in a domain. There are three main areas of application for ontologies according to [9]. They are for communication and knowledge sharing, logical inferencing or reasoning, and knowledge reuse. One language that can be employed in modeling and reasoning about context information is the OWL Web Ontology Language [10] which has received strong support from the academic, medical and commercial sectors.

```
<owl:Class rdf:ID="ContextEntity"/>
  <owl:Class rdf:ID="Location">
  <rdfs:subClassOf rdf:resource="#ContextEntity"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="locatedIn">
  <rdf:type rdf:resource="FunctionalProperty">
  <rdfs:domain rdf:resource="Room">
  <rdfs:range rdf:resource="xsd:double">
</owl:ObjectProperty> ...
<owl:ObjectProperty rdf:ID="food_intake">
  <rdf:type="owl:TransitiveProperty"/>
  <rdfs:domain rdf:resource="#Healthy"/>
  <rdfs:range rdf:resource="#Vegetable"/>
  <owl:inverseOf rdf:resource="#has_nutrient"/>
</owl:ObjectProperty> ...
```

Fig. 1. Context ontology using OWL

In our research, the context ontology is written in OWL as a collection of RDF triples, where each statement being in the form (subject, predicate, object), where subject and object are ontology's objects or individuals, and predicate is a property relation defined by the ontology. For instance, in the example application scenario

entities can be modeled using OWL as shown in Fig. 1. Mr. Kim is staying inside the room and cautiously taking healthy food such as vegetables. The use of OWL not only facilitates semantic interactions with context information but also improves the scalability of decision-support applications.

4.2 Context Modeling

Context modeling is the specification of all entities and relations between these entities which are needed to describe the context as a whole. The use of context ontologies can solve the problem of using proprietary representation schemes which hinder the interoperability of the different computation entities. Modeling context using an ontology-based approach allows us to describe contexts semantically in a way which is independent of programming language, underlying operating system or middleware.

The context ontologies consist of the upper ontology for the general concepts and domain-specific ontologies which apply to different subdomains. The upper ontology is fixed once it is defined and will be shared among different domains. The domain-specific ontologies are composed of a collection of low-level ontologies which define the details of the general concepts and their properties in each subdomain such as home domain, office domain or vehicle domain as shown in Fig. 2 [11].

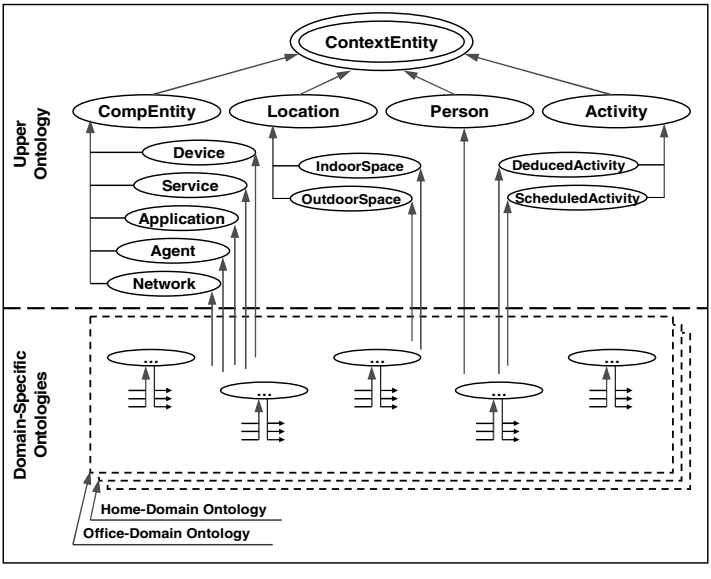


Fig. 2. Context ontology

4.3 Context Reasoning

Context reasoning refers to further deduction of previously implicit facts from explicitly given context information. Context reasoning is necessary in calculating high-level context information from a low-level sensor data. It is also necessary in checking and solving the inconsistencies in sensor data.

5 System Architecture

Based on the context model, we design the system architecture as shown in Fig. 3., which aims to provide an efficient infrastructure support for building context aware services in ubiquitous computing environments. A platform deploys the context-aware and service oriented architecture on the heterogeneous hardware and integrates software components with the web service technologies. As a result, the flexible system is capable of attending elder persons by providing appropriate services in an independent and efficient way.

The system architecture consists of the following components that act as independent service components. The system architecture consists of the following components that act as independent service components namely: context interpreter, context database, context event broker, web services adapter, context interface and access control, and virtual and physical Sensors.

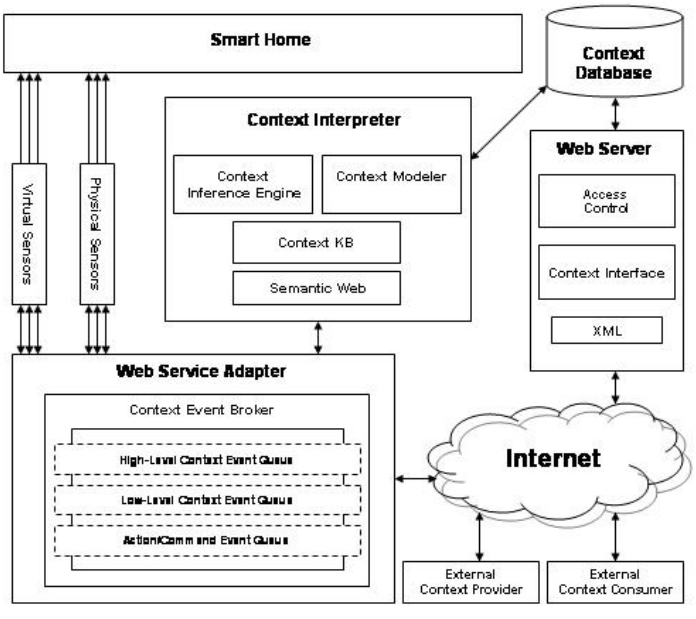


Fig. 3. System architecture

The context interpreter performs context processing which includes deriving high-level contexts from low-level contexts, querying context knowledge, maintaining consistency of context knowledge and resolving context conflicts. It consists of context inference engine, context modeler and context Knowledge Base (KB). The context inference engine provides deduced context based on direct contexts, detecting inconsistency and conflict in the context knowledge base. The context KB provides a set of API's for other service components to query, add, delete or modify context knowledge. It contains context ontology in sub-domain and their instances which can

be defined by users in case of specified contexts or acquired from various context providers in case of sensed contexts.

The web service adapter provides a gateway for agents to communicate with external context service. Each service can lookup and then bind to other services dynamically with a context repository. The external context providers and consumers feed and monitor the context events to provide appropriate services to the smart home environment.

In handling messages in XML, the Context Inference Engine uses the Context Modeler and related ontology to provide context input to the system. Accessing the services provided by the system is restricted by the access control. The component for access control maintains a list of all authorized users which contains their defined access control types. In the system, we implemented a rule-based approach for reasoning on contexts. A rule-based system allows developers to construct knowledge base consisting of rules and facts, and to deduce new facts using context inference engine. As a result, the decision-support system is able to provide appropriate services in a timely manner depending on the situation.

6 System Interface and Evaluation

The system was designed to integrate a variety of home appliances and sensors for the smart home environment consisting of mobile phones, desktop computer, laptop, web camera and sensors. It was developed using Microsoft Visual C# 2008 with context interpreter for construction and maintenance of an application or service. For the semantic web framework, it implements the Jena2-HP's Semantic Web Toolkit. Authorized users can retrieve information remotely via Internet-ready mobile devices.

The ontology for context modeling and reasoning was developed using OWL. Besides communications to sensors for vital signs monitoring and identification, e.g., wearable sensors and RFID readers, the gateway also manages a variety of smart devices that can be smart chair, smart table, etc. These smart appliances are controlled by computers and can communicate with the home gateway through the local area network. Shown in Fig. 4 are context-aware healthcare service interfaces that process and display patient's vital information through sensors. The raw data as categorized as sensed context are processed by context interpreter to provide useful context information. The system automatically updates the current context from sensors and performs context-aware reasoning using forward chaining rule inference.

We evaluated the applicability of context-aware service framework using a decision-support application for preventive healthcare by conducting a survey of the system to elicit responses from 35 random users, 15 selected health-care personnel, and 12 elders. The evaluation results are shown in Table 1 in which the criteria for evaluation were rated by respondents in a scale of 1-10. The results clearly showed that the user acceptance are lower compared to other criteria due to elders being hesitant and afraid to use new technologies but they show eagerness to learn and use the system with proper assistance and time spent in getting acquainted with the system.

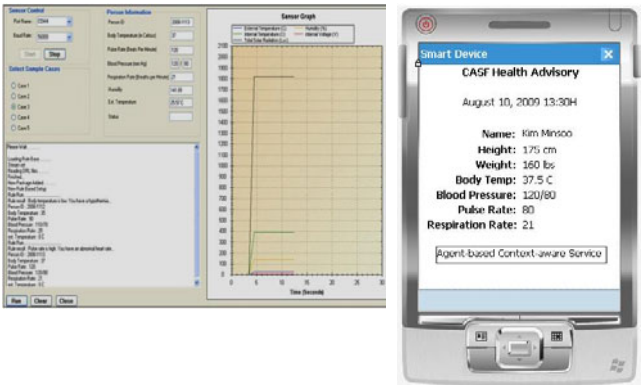


Fig. 4. Context-aware Healthcare Service

Table 1. Evaluation results

Criteria	Rating
User-friendliness	8.92
User acceptance	7.14
Applicability	9.31
Usefulness	9.15
Robustness	9.28
Reliability	8.93

7 Conclusion

This researched has implemented a decision-support system designed to facilitate preventive healthcare for elders. It has shown that the application of ontology can be used to support context modeling and reasoning by applying the general application areas of ontologies to the domain of context in ubiquitous computing environments. These areas include knowledge sharing, logical inferencing and knowledge reuse. The system provided a variety of services for the elders, as well as related services for healthcare personnel, family, and service providers. Connecting all caring services together in a context-aware service framework, the decision-support system integrates them to provide personalized services to elders.

The application of ontology for context information in ubiquitous computing environments lies in the interoperability of different devices. The current implementation deals with preventive care scenarios to elders in order to demonstrate all the system functionalities. For future studies, we will explore more complex cases such as handling conflicting data from multiple sensors.

Acknowledgment

This paper has been supported by the 2010 Hannam University Research Fund.

References

1. Dey, A., Salber, D., Abowd, G.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. In: Hum.–Comput. Interact (HCI), pp. 97–166 (2001)
2. Chen, G., Kotz, D.: A Survey of Context-Aware Mobile Computing Research. Technical Report. Dartmouth College (2000)
3. Chen, H., Finin, T.: An ontology for a context aware pervasive computing environment. In: IJCAI Workshop on Ontologies and Distributed Systems, Acapulco Mexico (2003)
4. Hong, J., Landay, J.: An infrastructure approach to context-aware computing. Human–Computer Interaction (2001)
5. Ranganathan, A., Campbell, R.: A middleware for context-aware agents in ubiquitous computing environments. In: ACM/IFIP/USENIX International Middleware Conference. Rio de Janeiro, Brazil (2003)
6. Horrocks, I.: DAMLC + OIL: a Reasonable Web Ontology Language. In: Proceedings of the Eighth International Conference on Extending Database Technology, Prague (2002)
7. Chen, G., Kotz, D.: Context Aggregation and Dissemination in Ubiquitous Computing Systems. Dartmouth College (2002)
8. Huang, Y., Garcia-Molina, H.: Publish/Subscribe in a Mobile Environment. In: Proceedings 2nd ACM International Workshop on Data Engineering for Wireless and Mobile Access (2002)
9. Gruninger, M., Lee, J.: Ontology – Applications and Designs. Communications of the ACM (2002)
10. Bechhofer, S., Harmelen, F., Hendler, J., Horrocks, I., et al.: OWL Web Ontology Language Reference. W3C Recommendation (2004)
11. Wang, X., Gu, T., Zhang, D., Pung, H.: Ontology Based Context Modeling and Reasoning using OWL. In: Second IEEE Annual Conference on Pervasive Computing and Communications Workshops (2004)

A Disaster Management Metamodel (DMM) Validated

Siti Hajar Othman and Ghassan Beydoun

School of Information Systems and Technology, Faculty of Informatics,
University of Wollongong, Wollongong NSW 2522, Australia
{sho492,beydoun}@uow.edu.au

Abstract. In this paper, we present the development and validation of a Disaster Management Metamodel, a language that we develop specific for describing disaster management domain, as a foundational component to create a decision support system to unify, facilitate and expedite access to disaster management expertise. The metamodel which consists of four views based on disaster management phases including Mitigation, Preparedness, Response and Recovery-phase classes is developed by using seven (7) steps of metamodel creation process. To check the expressiveness and the completeness aspect of the metamodel, we validate this representational metamodel by analysing a validation over ten well-known disaster management metamodels which are chosen based on a *Model Importance Factor* criteria. The paper presents the synthesis process, the resulting metamodel and its validation.

Keywords: Metamodel, Disaster Management, Modelling Language, Validation, Concept, Model.

1 Introduction

The increasing number of disasters recently, such as earthquakes, tsunamis, floods, bushfires, air crashes, epidemic, have posed a huge challenge not only to population at large, but also to public services and agencies tasked with activities relating to preventing and managing disaster responses. Recent failures can be easily identified in the management of the Swine-Flu (H1N1) pandemic hitting Australian shores in large numbers through cruise ships or in the devastating communication failures in the recent bushfires in Victoria (Australia). Many such failures are due to expertise not being available in a timely manner. This is partly due to inability to recognize and identify correct expertise, as it is often perceived as too tied to kinds of events (floods, bushfires, tsunamis, pandemic or earthquake). Potential of reusing expertise is often overlooked leading to catastrophic consequences. In this paper, we present an approach to unify DM knowledge to create a DM Decision Support System (DSS) that combines and matches different DM activities to suit the disaster on hand.

Disaster Management (DM) is a multidisciplinary endeavor and a very difficult domain to model. Moreover, DM often depends on various types of information systems such for modelling, simulation, visualization or management of geographical information, to support decision makers in all stages of a disaster [1]. Conceptual

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 111–125, 2010.

modelling of this domain is indeed a very challenging task especially to newcomers to the field. Our research which will remedy the situation through the introduction of a generic DM Metamodel (DMM) which can be used to manage this complexity of this domain through dividing all identified common concepts that exist in many DM models into four different views. In this paper, we clearly group concepts classes into four areas of concern: Mitigation, Preparedness, Response and Recovery classes.

The work is part of an effort to develop a Decision Support System (DSS) in DM. The system will be based on a Cased Based Reasoning approach. It will accept as input a query suitably expressed using concepts from the metamodel described in this paper, and outputs a model fragment that can be used by the decision maker as a suggestion to zoom in on plausible DM actions for the query input. We are currently developing the suitable input/output representations of the system. The rest of this paper is structured as follows: Section II describes the seven-step creation processes of developing the initial version of DMM. Section III presents the resultant of DMM which consists of four views based on DM phases including *Mitigation*, *Preparedness*, *Response* and *Recovery*-phase concepts of DMM. Section IV presents an evaluation of DMM over ten well-known DM models chosen based on MIF criteria and Section V concludes the paper with a discussion on our findings and future work.

2 Disaster Management Metamodel Creation Process

To construct the DMM, a set of common concepts to be used in the metamodel is first to be determined. A *concept*, the main components in a metamodel is an abstract object which represents an entity, action or a state [7]. Our identified DM concepts and their definitions are rooted in the existing literature related to DM models and metamodels. By capturing and collecting these frequently occurring concepts in any DM models or metamodels, relationships among all these concepts can be identified. The metamodel creation process is an iterative process with continuous refinement of new concept performed. To create the DMM, we adapt metamodelling steps that have been used in [3] and [8].

Step 1: ***Identifying models by using Model Importance Factor (MIF).*** Many disaster models have been developed worldwide. To identify a subset of most influential models as input for our metamodelling process, we formulate a new criterion as Model Importance Factor (MIF) shown in (1) to calculate a heuristic measure to compare the relevancy of various models. The top 10 most influential models [9-17] are used as input for Step 2. The rest are used for validation in order of most relevant. In developing this MIF, we adapt the idea of *Journal Impact Factor* measuring the frequency of which the average article in a journal has been cited in a particular year and we add additional weight to the size of the organization publishing the model. Our MIF will compare the impact of the models in the same domain. MIF is defined as follows:

$$\textbf{Model Importance Factor} = \frac{(T_{cited} * (E_{level} * P) * R_{coverage})}{((Y_{current} + 1) - Y_{published})} \quad (1)$$

T_{cited}	:	Model citation count; For a model appear in a publication without a citation, a default weight is used as follows: Research thesis is 10; Academic report is 15;
$Y_{published}$	:	The Year model is published;
$Y_{current}$	:	The current Year calculation is made; (2010 in our case)
E_{level}	:	Weight of Effort is allocated as follows: 0.1 for Individual; 0.2 for National Organization, 0.3 for International Organization;
$R_{coverage}$	:	An estimate of how applicable the model to the DMM development requirement, i.e. how much of the disaster management domain is covered.
P	:	The number of Participants involved in developing a model. This is estimated using number of co-authors of a model, or the size of committee involved in synthesizing the model.

Step 2: ***Extraction of general concepts in models identified in Step 1.*** During this step, disaster-specific concepts were omitted e.g.: *earthquake magnitude*, *tsunami warnings*, *fire danger index*, *Haiti earthquake victims* or *bushfire evacuation*. Chosen concepts are disaster type independent.

Step 3: ***Short-listing candidate definitions.*** During this step, a greater weight is given to sources with clearer definitions (in favor of those considered implicit definitions that can be subject to interpretation). Widespread occurrence of any particular DM definition is also taken into account leading to adopting a set of general concept grounded in commonly agreed meaning in DM community. A total of 137 concept definitions are short listed. All chosen concepts are represented through rectangular UML concept symbol in each four classes of our resultant metamodel in Fig. 1 - 4). E.g.: *Hazard Assessment* concept in Mitigation-phase class (Fig. 1).

Step 4: ***Reconciliation of definitions where possible.*** In choosing the common concept definition to be used, consistency with earlier choices is maintained. Further, if there is inconsistency between two or more sources occurs (especially because DM involved various kind of disaster), we choose the concept which has more coherent usage with the rest of the chosen concepts. As for an example the concept of *disaster* exist in many models we investigate. However, the definitions of the concept in each model are defined differently. Thus the reconciliation of this concept is demanded.

Step 5: ***Designation of concepts into 4 sets: Mitigation, Preparedness, Response and Recovery.*** Through studies to many extant disaster models, we observed that many disaster organizations, emergency groups and researchers organizing their DM activities in four disaster phases including *Mitigation*, *Preparedness, Response* and *Recovery*. *Mitigation* is a phase of which DM seeks to eliminate or reduce the impact of disasters themselves and/or to reduce the susceptibility and increase the resilience of the community subject to the impact of those hazards. *Preparedness,* the phase to establish arrangements, plans and provides education and information to prepare the community to deal effectively with disasters as they may eventuate. *Response* phase will activate preparedness arrangements and plans to put in place effective measures to deal with emergencies and disasters if and when they do occur and lastly *Recovery* will assist a community affected by an emergency or disaster in reconstruction of the physical infrastructure and restoration of emotional, social, economic and physical

well-being. Thus we designated DM concepts that we derived according to its DM phase respectively.

Step 6: ***Identification of relationships*** within and across *Mitigation, Preparedness, Response* and *Recovery* diagram and relationships interfacing the categories. Output of this step is the initial DMM (Fig. 1 - 4). For each subset of concepts, we identify relationships between concepts and produce a single diagram depicting the phase. For example, *EmergencyManagementTeam* and *Coordination* concepts, based on our observations, we set the relation of '*Requires*' to indicate the emergency management team requires coordination when they perform the rescue task during any emergency situation. Besides relationships created between the concepts from the same class, we also consider relationships exist between concepts of different phases.

Step 7: ***Validating the metamodel***. We use a '*Comparison to other Metamodels'* validation technique to evaluate our metamodel. Ten metamodels have been chosen based on MIF factor and shown in Table 1. In this paper, we present only two of the samples (Section 4.1-DOM and Section 4.2-CWML).

Table 1. Respective MIF values of models chosen to be used as a validation set of DM metamodels to evaluate the DMM (Step 7)

	10 Metamodels for validating the DMM (Step 7)	**MIF**	T_{cited}	$Y_{published}$	$D_{standard}$	P	R
A	Community Resilience as a Metaphor, Theory, Set of Capacities and Strategy for Disaster Readiness [18]	**2.03**	15	2001	0.3	15	0.3
B	Using SDI and Web-based System to Facilitate Disaster Management [19]	**1.35**	10	2005	0.3	9	0.3
C	A Framework for Modelling and Simulation for Emergency Response [5]	**1.29**	10	2004	0.3	10	0.3
D	Chaos, Crisis and Disaster Management: Strategic approach to Crisis Management in the Tourism Industry [20]	**1.00**	15	2002	0.2	10	0.3
E	Humanitarian Logistics in Disaster Relief Operations [21]	**0.82**	15	2000	0.2	10	0.3
F	Computer-based Model for Flood Evacuation Emergency Planning [22]	**0.24**	10	2006	0.1	4	0.3
G	Disaster mitigation: the concept of vulnerability revisited [23]	**0.19**	8	2008	0.1	7	0.1
H	OR/MS research in disaster operations management [24]	**0.12**	3	2008	0.1	4	0.3
I	Integrated Community-Based Disaster Management In Taiwan [25]	**0.10**	1	2008	0.2	5	0.3
J	Cyclone Warning Markup Language (CWML) [4]	**0.02**	2	2006	0.1	2	0.3

3 The Resultant Metamodel

This section presents our initial DMM. We first present our metamodel in four different diagrams to clearly group classes into four areas of concern: *Mitigation*-phase

(Fig. 1), *Preparedness*-phase (Fig. 2), *Response*-phase (Fig. 3) and *Recovery*-phase (Fig. 4) class. Each figure shows classes which refer to concepts that should exist during a corresponding phase of DM. The resultant metamodel contains the relationships among concepts and represents the semantic of the DM domain. As an example, the Response-phase (Fig. 3) has a central concept *ResponseOrganization* concept. An aggregation symbol (—◇) shows relationships between *ResponseOrganization* concepts and each of *Resource*, *EmergencyManagementTeam*, *EmergencyOperationCentre*, *EmergencyPlan, Aid* and *Rescue*. In any response phase of disaster, an organization requires resource, emergency team, centre to control coordination, emergency plan, aid and rescue tasks. Another relationship between concepts is association (denoted by symbol of (——)). For example, an association is shown between *EmergencyManagementTeam* and *ResponderTask* concepts. For example, a task of a response actor (person) is defined by the emergency management team. Another example, a *Resource* concept '*Requires*' *Deployment* concept, indicating that during any response phase, emergency resources such as rescue equipments, police transportation, fire equipments or medicine have to be deployed to help disaster victims.

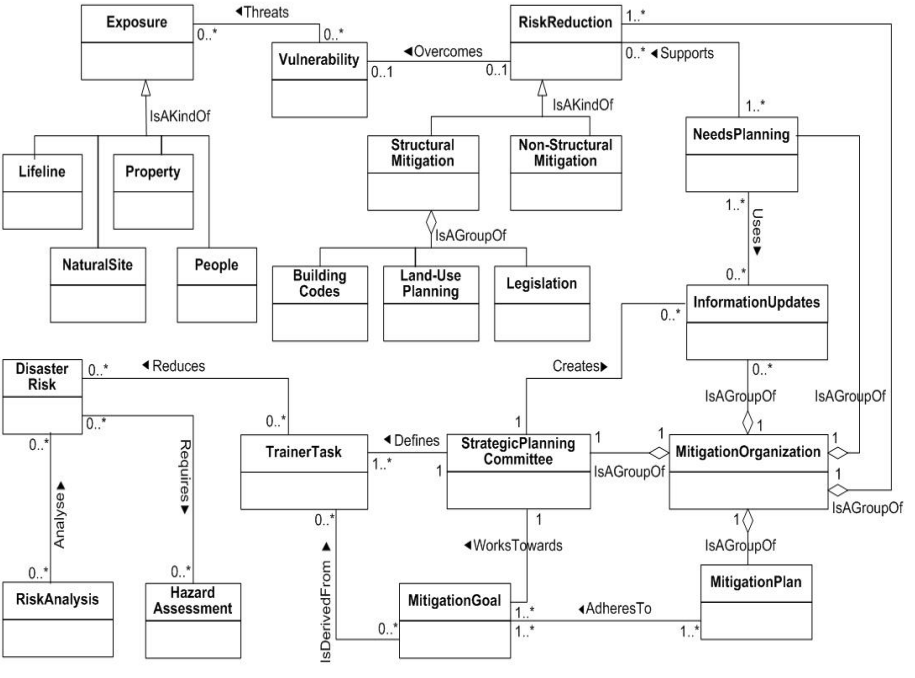


Fig. 1. *Mitigation*-phase concepts of DMM

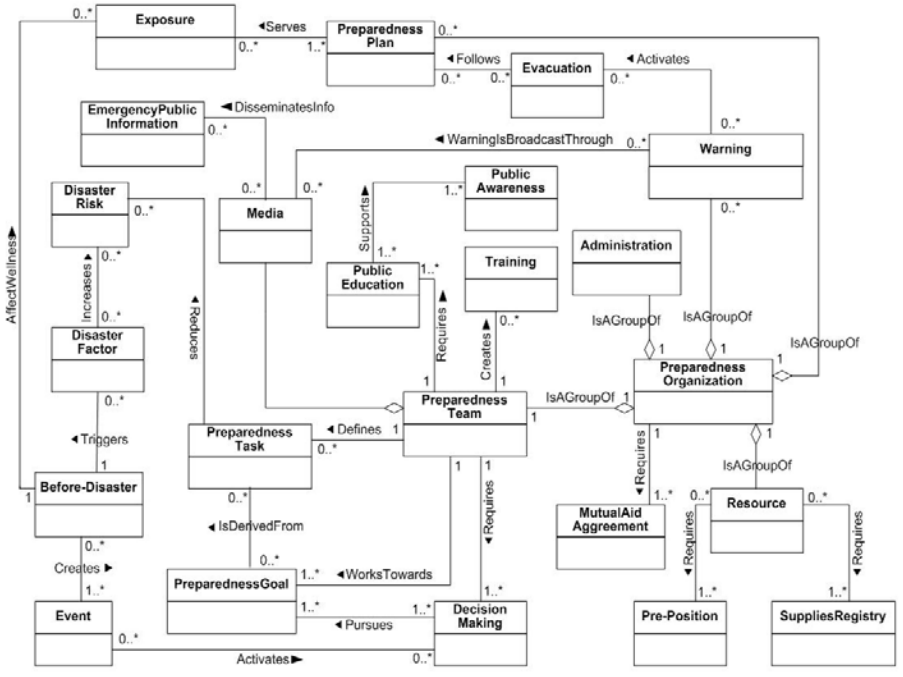


Fig. 2. *Preparedness*-phase concepts of DMM

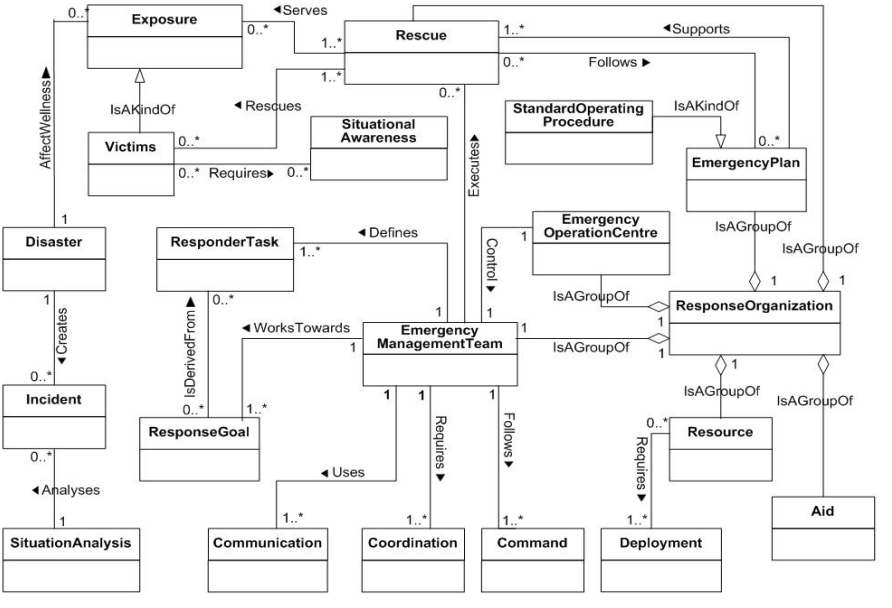


Fig. 3. *Response*-phase concepts of DMM

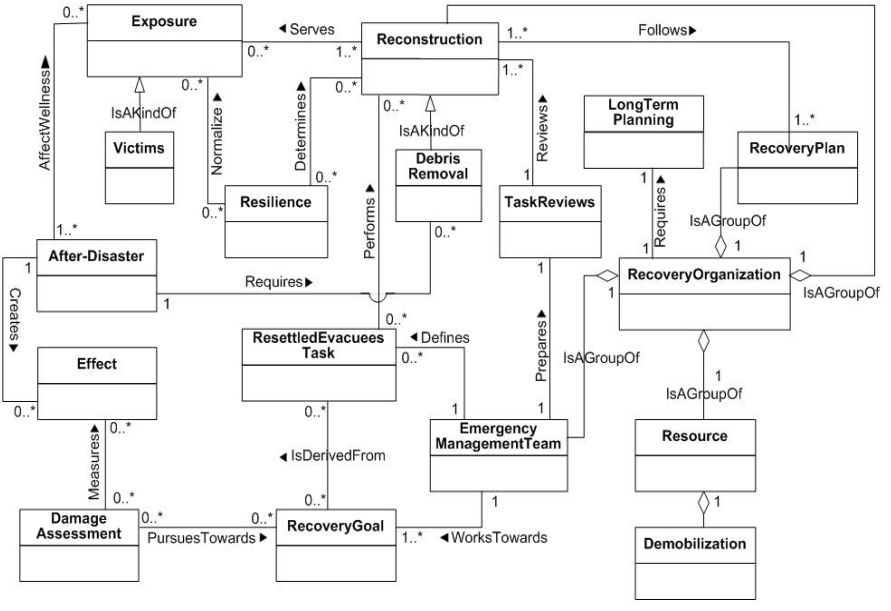


Fig. 4. *Recovery*-phase concepts of DMM

4 Validating the Metamodel

To satisfy requirements of generality, expressiveness and completeness aspects, DMM itself requires validation. *Conceptual metamodel validation* as discussed by Sargeant [26] is defined as a process in determining that the theories and assumptions underlying concepts in the metamodel are correct and the representation of metamodel of the problem entity and the structure of the metamodel, logic and causal relationships are *reasonable* for the intended purpose of the metamodel. Validation also determines that an agreement has been achieved among concepts in the metamodel against real data of

Table 2. Conceptual validation techniques of metamodel concepts [26]

Technique	Definition
Comparison against metamodels	Derived concepts of the developed metamodel are validated and being compared to concepts of other (valid) existing similar domain models or metamodels.
Multistage validation	Combination of three historical methods of rationalism, empiricism, and positive economics into a multistage process of validation.
Tracing	The behavior of different types of specific entities in the model is traced (followed) through the model to determine if the logic of the model is correct and if the necessary accuracy is obtained.
Face validity	Asking individuals knowledgeable about the domain application whether the model and/or its behavior are reasonable.

a domain. Commonly used validation techniques are shown in Table 2. We choose the first technique to apply in the rest of this section.

4.1 Validation against Disaster Operation Management Metamodel (DOM)

One of metamodel we use to validate DMM is The Disaster Operation Management Metamodel, DOM [24] (see Model H in Table 1). In this model, the focus is given more on the way we designate the sets of our metamodel concept. We identify that OR/MS model supports the idea of representing our metamodel into four main DM phases. Table 3 shows DOM activities and their respective phase. Hence, we continue our validation process with evaluating each concept which represents activities involved in each phase. We note that our DMM is missing some concepts include *RefugeeShelter* and *MassCasualtyManagement* in *Response*-phase class and *Insurance* concept in *Mitigation*-phase class. We therefore add these into our DMM.

Table 3. A list of some DOM concepts which support concepts in DMM

DOM concept (refining the corresponding DMM concept)	**DMM concept**
DOM Phase: MITIGATION	
Zoning and Land use controls	*Land-UsePlanning*
Building codes to improve disaster resistance	*BuildingCodes*
Preventives measures	*Non-/StructuralMitigation*
Controls on rebuilding after events	*StructuralMitigation*
Risk analysis to measure extreme hazards	*RiskAnalysis*
Insurance to reduce the financial impact	-Not supported- Add in Mitigation
DOM Phase: PREPAREDNESS	
Recruiting community volunteer groups	*VolunteerTask*
Emergency Planning	*EmergencyPlan*
Development of mutual aid agreement	*MutualAidAgreement*
Training both response personnel and concerned citizens	*Training*
Threat based public education	*PublicEducation*
Budgeting for and acquiring vehicles and equipment	*Administration*
Maintaining emergency supplies	*SuppliesRegistry*
Constructions of emergency operations center	*EmergencyOperationCentre*
Development of communication system	*Communication*
Conducting disaster exercises	*Training*
DOM Phase: RESPONSE	
Activating emergency operation plan	*EmergencyPlan*
Activating emergency operation center	*EmergencyOperationCentre*
Evacuation of threatened population	*Evacuation*
Opening shelters and provision of mass casualty	-Not supported- Add in Response
Emergency rescue and medical care	*Rescue*
Fire fighting	*ResponderTask*
Urban search and rescue	*Rescue*
Emergency infrastructure protection and lifeline recovery	*Reconstruction*

Table 3. (*continued*)

Fatality management	-Not supported- Add in Response
DOM Phase: RECOVERY	
Disaster debris cleanup	*DebrisRemoval*
Financial assistance to individual and governments	-Not supported- Add in Recovery
Sustained mass care for displaced human and animal	*Reconstruction*
Reburial of displaced human remains	-Not supported- Add in Recovery
Full restoration of lifeline services	*Reconstruction*
Mental health and pastoral care	-Not supported- Add in Recovery

4.2 Validation against the Cyclone Warning Mark Up Language Metamodel

A Cyclone Warning Mark Up Language (CWML) [4] is a standards-based language model which aims to maximize opportunities for interoperability for disaster cyclone

Table 4. CWML Support Concepts for DMM Preparedness-phase class

CWML concept	**CWML definition *(refining the more general DMM definition)***	**DMM concept**	**DMM definition**
Severe Weather Advisory	The top-level container element	*Emergency Operation Centre*	A facility, either static or mobile, from which the total operation or aspects of the emergency operation are managed.
Applicable Area	The applicable area of this threat	*Exposure*	People, property, systems or other elements present in hazard zones that are thereby subject to potential losses.
Warning	Contains a description of the type of the warning, the areas covered by the warning and the status of the warning.	*Warning*	The set of capacities needed to generate and disseminate timely and meaningful warning information to enable individuals, communities and organizations threatened by a hazard to prepare and to act appropriately and in sufficient time to reduce the possibility of harm or loss.
Action	The action that people should take for the severe weather event	*Prepared-nessTask*	A preparedness task defined by the *Preparedness Team.*
Threat	Provides an analysis of the threat caused by the severe weather event. This analysis will usually include a plain language description and a set of detailed predictions based on different threat factors.	*Disaster Factor*	An event, danger or occurrence of something that can contribute to the cause of disaster.

Table 4. (*continued*)

Media	Encapsulates the manner of broadcast and the usage of the warning signal.	*Media*	A communication channel through which news, education, data, information or warning messages are disseminated. Media includes every broadcasting and narrowcasting medium such as newspapers, magazines, TV, radio, billboards, direct mail, telephone, fax, and internet.
Flood	A description that characterizes expected flooding	*Event*	An incident or situation, which occurs in a particular place during a particular interval of time.

advices and is developed to define a structure of semantic data models for cyclone warning. This model particularly concentrates on preparedness phase since this model is being developed for warning purposes. Hence the focus of our validation is given more to *Preparedness* and *Mitigation*-phase class concepts. Table 4 illustrates concepts used by CWML compared to DM metamodel concepts with respective

Table 5. List of new added concepts after the validation process

New Concept	Phase	Definition
Insurance	Mitigation	A policy that is designed to provide an insurance alternative to disaster assistance to meet the escalating costs of repairing damage to properties and their contents caused by disasters.
Monitoring	Preparedness	The observation, measurement and valuation of disaster situation progress in order to identify change.
AidAgency	Preparedness	An organization dedicated to distributing aid includes within government, between governments as multilateral donors or private voluntary organizations.
Information Management	Response	The processes that collect, analyse, format and transmit data and information during an incident of disaster.
Refugee-Shelter	Response	An accommodation provided over an extended period of days, weeks or months for individuals or families affected by an emergency.
Mass-Casualty	Response	A coherent and interrelated set of established procedures, policies, and plans that contribute to the shared objectives
Management		of optimizing the baseline capacity to deal with patient populations expected in a mass casualty incident, and efficiently increasing this capacity during the response to a mass casualty incident.
FoodAid	Response	Assistance rendered on an organized basis, either free or on concessional terms, to provide food to a population group, community or country suffering from food shortage or insufficient development

Table 5. (*continued*)

MedicalAid	Response	Form of aid in types of medical supplies such as medicine, emergency first aid, healthcare equipment or other emergency health supplies to help assist people who are injured and suffered after a disaster hit.
Economic Restoration	Recovery	A response and recovery action which actively support the recovery of business, industry and economic structure.
Financial Assistance	Recovery	A provincial cost-sharing program with local government and private sector claimants based on provincial legislation provided to emergency affected persons, communities or organizations to assist their recovery from an emergency
Mental-Health Recovery	Recovery	A program that provide short-term, in-person, disaster-oriented, emotional support and problem solving assistance in a variety of settings for individuals and families who are attempting to deal with their fears and other negative psychological after-effects of a major disaster or large-scale emergency such as post-traumatic stress disorders, depressive or anxiety disorders, somatic complaints and general mental morbidity that disrupts the normal functioning of a community.

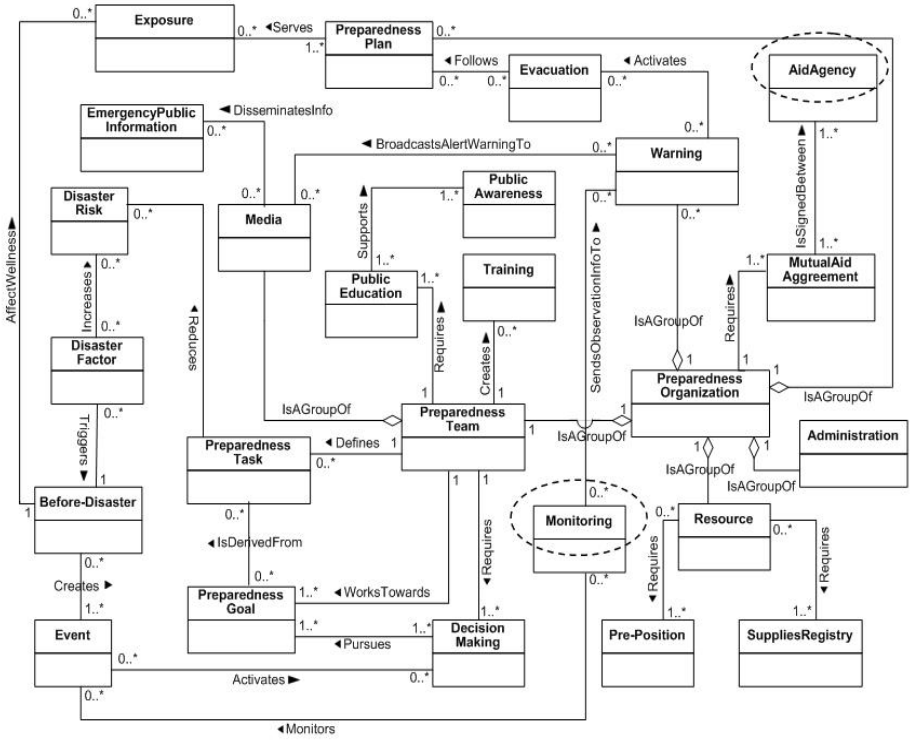


Fig. 5. A validated version of *Preparedness*-phase of DMM with new concepts shown

definition. Assessments we made to all CWML concepts prove that DMM manage to support almost all CWML concepts except for one, a '*Monitoring*' concept. Therefore, we add this concept into *Preparedness*-phase of our DMM.

We do not show the details of the rest of the validation against the 8 remaining metamodels due to space constraints. However, as a result of the complete validation against the 10 metamodels (shown in Table 1), we identify in total eleven (11) new added concepts and its definition (see Table 5), no deleted previous concepts and we add new six (6) relationships among the concepts. We also revise all existing DMM concepts and their relationships. For example, based on our observation against Model E (from Table 1), we review one relationship, '*Determines*' in *Recovery*-phase concepts of DMM (Fig. 4). We rename it to '*Supports*', to show reconstruction operation are actually supports and not determines the resilience from disaster.

We add all new concepts into DMM classes, linking them with new/existing concepts as required. The new DMM is shown in Fig. 5 to 7 with the new concepts highlighted (The *Mitigation*-phase of our DMM only gained the concept *Insurance* and its new diagram is omitted due to space constraints).

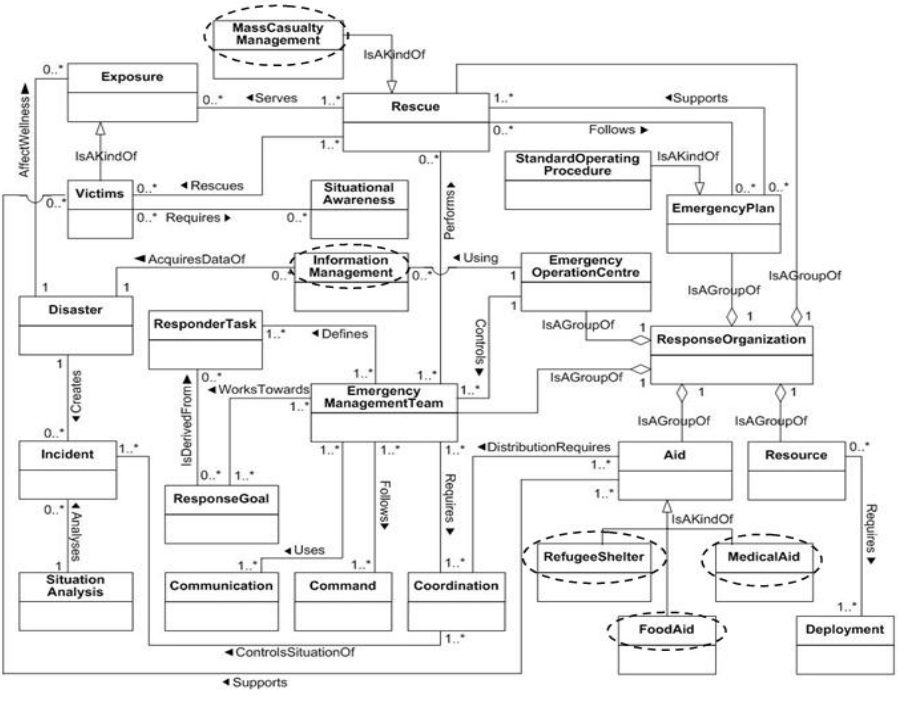


Fig. 6. A validated version of *Response*-phase concepts of DMM with new concepts shown

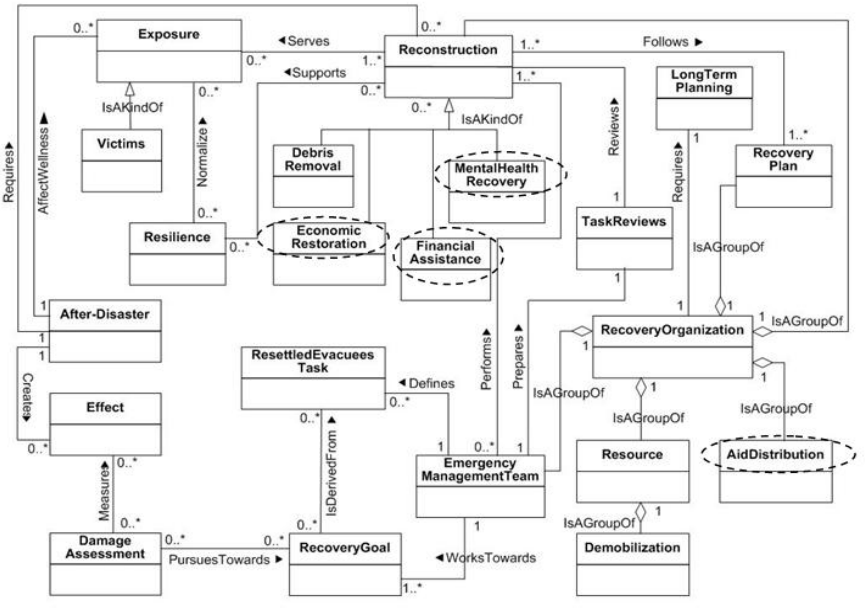


Fig. 7. A validated version of *Recovery*-phase concepts of DMM with new concepts shown

5 Conclusion and Future Works

This paper presents the development and validation of a Disaster Management Metamodel. The metamodel presented is intended to become a foundational component to create a decision support system to unify, facilitate and expedite access to disaster management expertise. It can also serve as an effective tool to determine the completeness of any DM solution. It will also allow interoperability of DM solutions and effective transfer of knowledge across international boundaries. We have presented the metamodel in a familiar format, UML, to increase its easy use and broaden its appeal. In the synthesis of our metamodel, we have collected 30 disaster models. In the work shown in this paper, we used 20 of those to validate our metamodel. The ability to represent key DM concepts using our metamodel is a preliminary evidence of the feasibility of a generic metamodel in DM. This is unlike most previous attempts in this area which have narrowed their focus on specific types of disasters. We will use the remaining 10 metamodels to further refine our metamodel. Following this validation, we will create a repository of DM knowledge expressed and engineered using our metamodel. In other words, DM actions will be represented as refinement of our metamodel concepts. This will be the first step to develop a DSS which assists in formulating required DM approach based on the disaster event that is provided as input to the system. The DSS itself will be based on a Cased Based Reasoning approach. It will accept as input a query suitably expressed using concepts from the metamodel, and outputs a model fragment that can be used by the decision maker as a

suggestion to zoom in on possible DM actions. We are currently developing the suitable input/output representations of the system.

References

1. Sotoodeh, M., Kruchten, P.: An Ontological Approach to Conceptual Modeling of Disaster Management. In: 2008 2nd Annual IEEE Systems Conference, pp. 1–4 (2008)
2. Kleijnen, J.P.C., Sargent, R.G.: A Methodology for Fitting and Validating Metamodels in Simulation. European Journal of Operational Research 120, 14–29 (2000)
3. Beydoun, G., Low, G., Henderson-Sellers, B., Mouraditis, H., Sanz, J.J.G., Pavon, J., Gonzales-Perez, C.: FAML: A Generic Metamodel for MAS Development. IEEE Transactions on Software Engineering 35(6), 841–863 (2009)
4. Sun, S., Iannella, R., et al.: Cyclone Warning Markup Language (CWML), Technical report, National ICT Australia, NICTA (2006)
5. Sanjay, J., McLean, C.: A Framework for Modeling and Simulation for Emergency Response. Simulation Conference 1, 1068–1076 (2003)
6. Nissen, H.W., Jeusfeld, M.A., Jarke, M., Zemanek, G.V., Huber, H.: Managing Multiple Requirements Perspectives With Metamodels. IEEE Software 13, 37–48 (1996)
7. Sowa, J.F.: Conceptual Structures: Information Processing In Mind and Machine. Addison-Wesley Longman Publishing Co., Inc., Amsterdam (1984)
8. Beydoun, G., Low, G., Mouraditis, H., Henderson-Sellers, B.: A Security-Aware Metamodel For Multi-Agent Systems. Journal of Information and Software Technology 51(5), 832–845 (2009)
9. Emergency Management Australia (EMA): Emergency Managemen. Australia Concepts and Principles (Manual 1), Technical Report (2004)
10. World Health Organization (WHO): Concepts in Emergency Management, The basis of EHA Training Programmes in WPRO
11. W3C Incubator Group: Emergency Information Interoperability Frameworks (2008)
12. Modoc County Disaster Council, California, USA: Emergency Operations Plan. California (2000)
13. Russo, E.E.R., Raposo, A.B., et al.: A Metamodel for Configuring Collaborative Virtual Workspaces: Application in Disaster Management of Oil & Gas, Technical Report (2006)
14. Cutter, S.L., Barnes, L., Berry, M., Burton, C., et al.: A place-based model for understanding community resilience to natural disasters. Global Environmental Change 18 (2008)
15. Kruchten, P., Monu, et al.: A Conceptual Model Of Disasters Encompassing Multiple Stakeholder Domains. International Journal of Emergency Management 5, 25–56 (2008)
16. Benaben, F., Hanachi, C., Lauras, M., Couget, P., Chapurlat, V.: A Metamodel and its Ontology to Guide Crisis Characterization and its Collaborative Management. In: Proceedings of the 5th International ISCRAM Conference Washington, USA, pp. 189–196 (2008)
17. Asghar, S., Alahakoon, D., Churilov, L.: A Comprehensive Conceptual Model for Disaster Management. Journal of Humanitarian Assistance (2006)
18. Norris, F.H., Stevens, S.P., et al.: Community Resilience as a Metaphor, Theory, Set of Capacities and Strategy for Disaster Readiness. American Journal of Community Psychology 41
19. Mansourian, A., Rajabifard, A., Valadan Zoej, M.J., Williamson, I.: Using SDI and web-based system to facilitate disaster management. Computers & Geosciences 32 (2006)

20. Ritchie, B.W.: Chaos, crises and disasters: a strategic approach to crisis management in the tourism industry. Tourism Management 25, 669–683 (2004)
21. Kovacs, G., Spens, K.M.: Humanitarian Logistics In Disaster Relief Operations. International Journal of Physical Distribution & Logistics Management 37, 99–114 (2007)
22. Slobodan, S., Sajjad, A.: Computer-based Model for Flood Evacuation Emergency Planning. Natural Hazards 34, 25–51 (2005)
23. Weichselgartner, J.: Disaster mitigation: the concept of vulnerability revisited. Disaster Prevention and Management 10, 85–94 (2001)
24. Altay, N., Green Iii, W.G.: OR/MS Research in Disaster Operations Management. European Journal of Operational Research 175, 475–493 (2006)
25. Chen, L.C., Liu, Y.C., Chan, K.C.: Integrated Community-Based Disaster Management Program in Taiwan: A Case Study of Shang-An Village. Natural Hazards 37, 209–223 (2006)
26. Sargent, R.G.: Verification and Validation of Simulation Models. In: Proceedings of the 37th Conference on Winter Simulation. Winter Simulation Conference, Orlando, Florida (2005)

Another Investigation of an Interontologia between Chinese Lexical Systems and Roget's Thesaurus

Sang-Rak Kim, Jae-Gun Yang, and Jae-Hak J. Bae*

School of Computer Engineering & Information Technology, University of Ulsan, Ulsan, Republic of Korea
shem0304@gmail.com, {jgyang,jhjbae}@ulsan.ac.kr

Abstract. The present study presents the lexical category relevancy analysis of the Thousand-Character Text and Chinese radicals, to Roget's thesaurus. According to the comparison of the Thousand-Character Text and Roget's thesaurus, most of the 39 sections of Roget's thesaurus are relevant to Chinese characters in the Thousand-Character Text. The correlation coefficient is around 0.90. In the case of Chinese radicals, 30 sections of Roget's thesaurus are relevant to the radicals. The correlation coefficient is around 0.85, showing considerable relevancy between Chinese radicals and sections of Roget's thesaurus, as well.

Keywords: Thousand-Character Text, Chinese Radicals, Roget's Thesaurus, Ontology, Interontologia.

1 Introduction

With the development of the Internet, the volume of information is now incomparable with that in the past. In order to manage such a large amount of information, we need standardized classification systems. A standardized classification system can be created based on human cognitive ability to classify things. Everything in the world has its own unique characteristics, by which it is classified into specific categories, and we understand things more easily by associating them to related categories. In this way, we simplify information processing and understand perceived things better by classifying them systematically according to their characteristic.

The lexical classification system covered in this study can be divided into various types according to the use of words or information. Examples of application are in the areas of artificial intelligence, computational linguistics and information communication include information search, knowledge management, information system design, ontology building, machine translation, and dictionary compilation. There are also implemented lexical classification systems related to the vocabulary resources of natural languages such as Roget's thesaurus [1], WordNet [2], Lexical FreeNet [3], Kadokawa thesaurus [4], and EDR [5]. Cases of ontology building include KR Ontology [6], CYC Ontology [7], Mikrokosmos Ontology[8], SENSUS Ontology [9] and

* Corresponding author.

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 126–134, 2010.

HowNet[10], and there are business applications of ontology such as Enterprise Ontology[11], UMLS[12], UNSPSC[13], RosettaNet[14], ISO 2788[15] and ANSI Z39.19[16].

Lexical classification is concept classification by nature. Lexical classification systems mentioned above suggest that there are various concept classification systems today. It is said that people have the same cognition, memory, causal analysis, categorization, and reasoning process. They assume that if there is any difference, it is not from difference in cognitive process but from difference in culture or education [17]. As mentioned above, in the current situation that various concept classification systems are being used in different application areas, it is keenly required to interlock concept classification systems and intelligent information systems. In response to the demand, research is being made on ontology mapping, merge and integration, and semantic integration [18, 19]. A main research method is the utilization of shared ontology or finding mapping in ontological features.

However, if there is a general concept classification system (*interontologia*) [20] as a reference classification system, through which it will become more systematic and easier to integrate concept classification systems semantically. Thus, as a case study on general concept classification system, the present study examines the relevancy of lexical categorization between the Thousand-Character Text [21, 22], which is a representative Eastern classic, and Roget's thesaurus[1], which is a famous Western classified lexicon. In addition to this, we also investigate the relevancy between Chinese radicals and the thesaurus. Through this study, we analyze similarities between the two in categorization and classification.

2 Lexical Ontologies: The Thousand-Character Text, Chinese Radicals, and Roget's Thesaurus

The Thousand-Character Text (千字文) was written by Zhou Xingsi (周興嗣) by order of Emperor Wu (武帝) in the Liang (梁) Dynasty of China in around the 6th century, and transmitted and distributed to Korea early in ancient times. This is a representative classical Chinese textbook used widely to teach children. The oldest Thousand-Character Text annotated with pronunciation and meaning in Korean is the version of Han Seok-Bong published in 1583. There is also a record on an earlier version published in Gwangju in 1575. The Thousand-Character Text is old four-character verse composed of a total of 250 four-character phrases or 125 couplets and its materials are Chinese history, culture, etc. [21, 22].

Roget's thesaurus [1] was first published in 1852 by English surgeon Peter Mark Roget. This is the first synonym/antonym dictionary. Roget's thesaurus is not in meaningless alphabetical order. The thesaurus classifies lexical knowledge systematically. The top hierarchy is composed of 6 classes, under which are divisions. Each division is again subdivided into sections. In each hierarchy is unique entry information, and at the end of the hierarchical structure are listed a total of 1044 categories. Each category has a list of synonyms by part of speech. On the other hand, if a specific word in the list of synonyms refers to another category, the reference is expressed in the form of "Vocabulary &c. (Entry word) Entry number."

There is a study on lexical classification systems in Korean representative classics such as the Thousand-Character Text, Yuhap (類合) and Hunmongjahoi(訓蒙字會) [22]. In the study, they argued that the Thousand-Character Text is structured well and has a clear system. In addition, they emphasized the accuracy of its classification system that does not allow even a repetition of the same character among the 1000 characters. Largely according to semantic paragraph (content), they classified Thousand-Character Text as follows: astronomy, nature, royal task, moral training, loyalty and filial piety, virtuous conducts, five moral disciplines, humanity and justice, palace, meritorious retainers, feudal lords, topography, agriculture, mathematics, quiet life, comfort, miscellaneous affairs, skills, admonition, etc. They concluded that in presenting Chinese characters by semantic paragraph, the Thousand-Character Text arranges basic Chinese characters appropriately and is outstanding in terms of lexical system and the perception of basic Chinese characters.

Chinese radicals are index keys or classifiers which are used for organizing entries in Chinese dictionary. Hàn dynasty(漢朝) scholar Xǔ Shèn(許慎) categorized all the Chinese characters with a system of 540 graphic elements that was called bùshǒu (部首). The elements are character components and denote some common semantic or phonetic characteristics. Chinese lexicographers had continued to refine this system for indexing Chinese characters. The number of Chinese radicals was reduced to 214 in the dictionary Zìhuì(字彙) in 1615. The set of radicals became standard and is still used in Chinese dictionaries today [23].

3 Relevancy Analysis

This study has conducted analysis on concept relevancy through 5 steps as follows.

Step 1. Building Master Databases: In this step, we sort out Chinese characters from the Thousand-Character Text, and words from Roget's thesaurus. Then build the master databases for the characters in the Thousand-Character Text, for the radicals in Chinese Radicals, and for the words in Roget's thesaurus, respectively.

Step 2. Identifying Meanings of Chinese Characters in English: In this step, we translate the meaning(s) of each Chinese character in the Thousand-Character Text and Chinese radicals into English words. Then we keep the translation in a database, referring to Chinese-English dictionary Kingsoft2008[24], Classical Chinese Character Frequency List[25], YellowBridge[26], and CHINAKNOWLEDGE[27].

Step 3. Filling Fields in Master Databases with English Equivalents: In this step, we determine an English equivalent for each Chinese character in the Thousand-Character Text and for each Chinese radical. And then fill the fields with the words in corresponding master databases.

Step 4. Mapping Chinese Characters and Radicals to Roget's Thesaurus Categories: In this step, we map English words for Chinese characters in the Thousand-Character Text and for Chinese radicals, into categories of Roget's thesaurus. And then keep the category information in the prearranged fields in the master databases.

Step 5. Analyzing Mapping Results: Finally, we analyze each relevancy of Thousand-Character Text and Chinese Radicals with respect to Roget's thesaurus using graphs and correlation coefficients based on the mapping data.

4 Analysis Results

4.1 The Thousand-Character Text and Roget's Thesaurus

Among the 1,044 categories of Roget's thesaurus, 424 categories have one or more corresponding Chinese characters while 620 categories do not have any corresponding Chinese characters. We also have analyzed the mapping on the section level of Roget's thesaurus, which are higher categories in the hierarchy of the thesaurus. Table 1 compares the mapping on the category level with the one on the section level of Roget's thesaurus.

Table 1. Results of mapping by classification level (Chinese characters of Thousand-Character Text)

Level	Number of mapped entries	Number of unmapped entries	Total	Mapping rate (%)
Roget's Category	424	620	1,044	41
Roget's Section	38	1	39	97

Fig. 1 shows the correspondence between Chinese characters in the Thousand-Character Text and Roget's thesaurus on the section level. We can see that the number of Chinese characters changes as the number of Roget's thesaurus categories does. We have obtained the correlation coefficient r_{xy} for the association between the number of Chinese characters in the Thousand-Character Text and Roget's thesaurus categories on the section level.

$$r_{xy} = \frac{N\sum XY - (\sum X)(\sum Y)}{\sqrt{\left[N\sum X^2 - (\sum X)^2\right]\left[N\sum Y^2 - (\sum Y)^2\right]}} \tag{1}$$

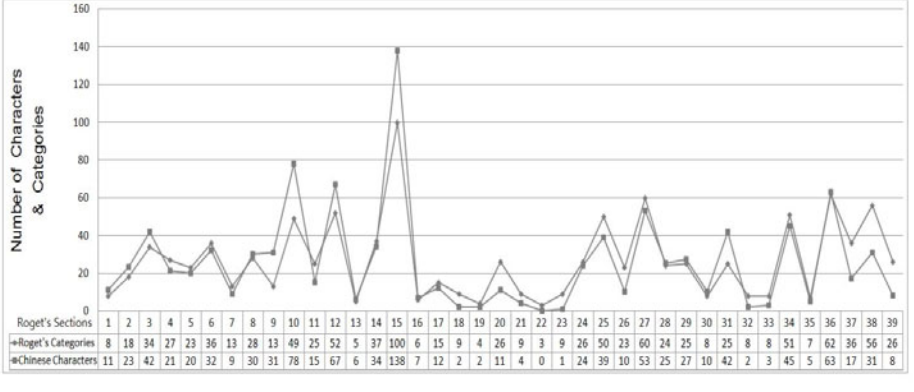


Fig. 1. Correspondence between the Thousand-Character Text and Roget's thesaurus on the section level

In the correlation analysis, the number of Roget categories is defined as X, the number of mapped Chinese characters as Y, and the number of Roget sections as N. For Equation (1), the values of variables are as follows.

$$\sum X_i = 1044,\ \sum Y_i = 1000,\ \sum X_i Y_i = 46038,\ \sum X_i^2 = 44474,\ \sum Y_i^2 = 53164$$

If these values are substituted for the variables in Equation (1), we obtain $r_{xy} = 0.90$, showing quite a high correlation between the Thousand-Character Text and Roget's thesaurus on the section level.

4.2 The Chinese Radicals and Roget's Thesaurus

We have also analyzed the mapping results between Chinese radicals and sections of the Roget's thesaurus. Table 2 shows the mapping rate on the section level of Roget's thesaurus. Table 3 shows the results. Among a total of 39 sections in Roget's thesaurus, three do not have any corresponding radicals, but 36 sections have one or more. The correlation coefficient is as high as 85%.

Table 2. Results of mapping by classification level (Chinese radicals)

Level	Number of mapped entries	Number of unmapped entries	Total	Mapping rate (%)
Roget's Sections	36	3	39	92

Fig. 2 and Table 3 shows the correspondence between Chinese radicals and Roget's thesaurus on the section level. We can see that the number of Chinese radicals changes as the number of Roget's thesaurus categories does.

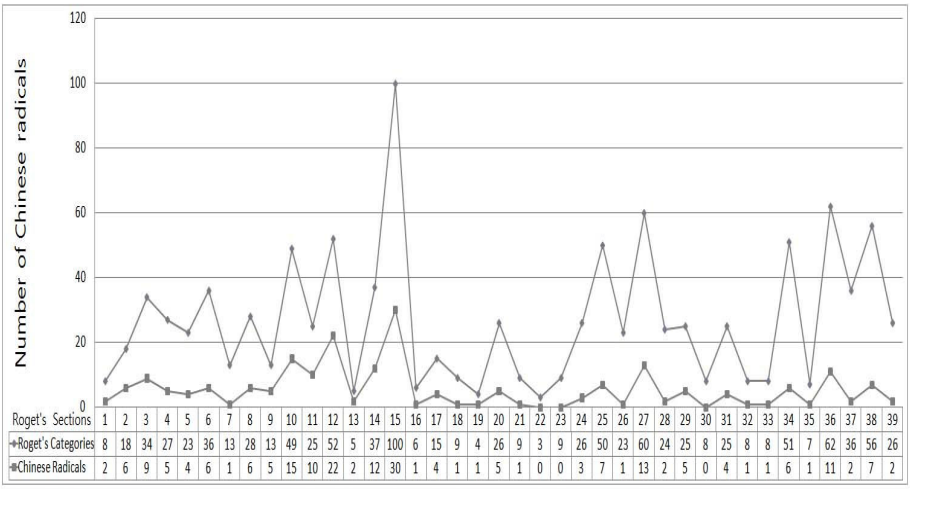


Fig. 2. Correspondence between Chinese radicals and Roget's thesaurus on the section level

In the correlation analysis, the number of Roget categories is defined as *X*, the number of mapped Chinese radicals as *Y*, and the number of Roget sections as *N*. For Equation (1), the values of variables are as follows.

$$\sum X_i = 1044,\ \sum Y_i = 214,\ \sum X_i Y_i = 9951,\ \sum X_i^2 = 44474,\ \sum Y_i^2 = 2652$$

If these values are substituted for the variables in Equation (1), we obtain $r_{xy} = 0.85$, showing a relatively high correlation between Chinese radicals and Roget's thesaurus on the section level.

Table 3. Mapping from Chinese radicals to Roget's thesaurus on the section level

No	Class	Roget's Sections	Chinese Radicals	Count
01	1	existence	立, 身	2
02	1	relation	韋, 方, 己, 比, 自, 血	6
03	1	quantity	寸, 丿, 大, 小, 尢, 支, 片, 而, 頁	9
04	1	order	鬥, 冂, 丨, 乙, 氏	5
05	1	number	鼎, 二, 又, 疋	4
06	1	time	夕, 子, 幺, 老, 辰, 長	6
07	1	change	艮	1
08	1	causation	力, 父, 虍, 龍, 屮, 爿	6
09	2	space in general	凵, 匚, 斗, 臼, 鬲	5
10	2	dimensions	亠, 勹, 匸, 厂, 口, 宀, 巾, 广, 毛, 瓦, 衣, 貝, 足, 高, 髟	15
11	2	form	丿, 刀, 彐, 彡, 穴, 角, 谷, 豆, 門, 齊	10
12	2	motion	儿, 入, 口, 夂, 戶, 瓜, 肉, 至, 舟, 行, 襾, 走, 辵, 酉, 飛, 食, 鬯, 魚, 黍, 皿, 缶, 癶	22
13	3	matter in general	山, 月	2
14	3	inorganic matter	土, 川, 气, 水, 田, 米, 糸, 雨, 革, 風, 骨, 麥	12
15	3	organic matter	一, 丶, 尸, 歹, 火, 甘, 白, 目, 羊, 耳, 色, 虫, 豕, 赤, 辛, 隹, 面, 韭, 音, 香, 馬, 鳥, 鹵, 鹿, 黃, 黑, 黽, 鼓, 龜, 龠	30
16	4	operations of intellect in general	示	1
17	4	precursory conditions and operations	几, 見, 鼻, 釆	4

Table 3. (*continued*)

No	Class	Roget's Sections	Chinese Radicals	Count
18	4	materials for reasoning	耒	1
19	4	reasoning processes	舛	1
20	4	results of reasoning	玄, 首, 艸, 士, 青	5
21	4	extension of thought	爻	1
22	4	creative thought		0
23	4	nature of ideas communicated		0
24	4	modes of communication	厶, 无, 毋	3
25	4	means of communicating ideas	丶, 文, 曰, 聿, 舌, 言	6
26	5	volition in general	鼠	1
27	5	prospective volition	冖, 匕, 工, 廾, 弋, 斤, 欠, 牙, 用, 疒, 石, 肉, 金	13
28	5	voluntary action	生, 彳	2
29	5	antagonism	戈, 殳, 矛, 矢, 皐	5
30	5	results of voluntary action		0
31	5	general intersocial volition	臣, 隶, 邑, 里	4
32	5	special intersocial volition	止	1
33	5	conditional intersocial volition	卩	1
34	5	possessive relations	皮, 网, 車, 攴, 手, 爪	6
35	6	affections in general	禾	1
36	6	personal affections	夂, 夊, 八, 十, 日, 牛, 人, 玉, 羽, 豸, 辮, 齒	12
37	6	sympathetic affections	弓, 心	2
38	6	moral affections	麻, 干, 非, 女, 木, 竹, 犬	7
39	6	religious affections	卜, 鬼	2

5 Conclusions and Future Research

The present study has examined concept relevancy between the Thousand-Character Text and Roget's thesaurus. Moreover, we have also conducted analysis on concept relevancy between Chinese Radicals and Roget's thesaurus. From the result of our experiment, we may say that there is an *interontologia* behind Roget's thesaurus and the Thousand-Character Text, or Chinese radicals.

Tasks for future research include: (1) complementing omitted parts in mapping of Chinese characters in the Thousand-Character Text to Roget's thesaurus categories with the 1800 commonly used Chinese characters in Korea and comparing the results; and (2) analyzing difference between comparison of the Thousand-Character Text and Roget's thesaurus on the category and section levels and (3) trimming 214 Chinese radicals into 100. From these studies, we can expect to have a set of Chinese characters for a refined lexical knowledge classification system. Lastly, based on the character set, we will develop a new lexical category system applicable to knowledge classification.

Acknowledgments. This work was supported by the Korea Research Foundation Grant funded by the Korean Government. (KRF-2008-313-H00009)

References

1. Roget's Thesauri, http://www.bartleby.com/thesauri/
2. WordNet, http://wordnet.princeton.edu/
3. Lexical FreeNet, http://www.cinfn.com/doc/
4. Ohno, S., Hamanishi, M.: New Synonyms Dictionary, Kadogawa Shoten, Tokyo (1981) (Written in Japanese)
5. The EDR Electronic Dictionary, http://www2.nict.go.jp/r/r312/EDR/index.html
6. KR Ontology, http://www.jfsowa.com/ontology/
7. CYC Ontology, http://www.cyc.com/
8. Mikrokosmos Ontology, http://crl.nmsu.edu/Research/Projects/mikro/htmls/ontology-htmls/onto.index.html
9. SENSUS Ontology, http://www.isi.edu/natural-language/projects/ONTOLOGIES.html
10. HowNet, http://www.keenage.com/html/e_index.html
11. Enterprise Ontology, http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html
12. UMLS, http://www.nlm.nih.gov/research/umls/
13. UNSPSC, http://www.unspsc.org/
14. RosettaNet, http://www.rosettanet.org
15. ISO 2788, http://www.collectionscanada.gc.ca/iso/tc46sc9/standard/2788e.htm
16. ANSI Z39.19, http://www.niso.org/standards/resources/Z39-19-2005.pdf
17. Nisbett, R.E.: The Geography of Thought: How Asians and Westerners Think Differently... and Why. Simon & Schuster, New York (2004)
18. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. The Knowledge Engineering Review 18(1), 1–31 (2003)
19. Noy, N.F.: Semantic Integration: A Survey of Ontology-Based Approaches. SIGMOD Record 33(4), 65–70 (2004)

20. Kim, S.-R., Yang, J.-G., Bae, J.-H.J.: An Investigation of an Interontologia: Comparison of the Thousand-Character Text and Roget's Thesaurus. In: Li, W., Mollá-Aliod, D. (eds.) ICCPOL 2009. LNCS (LNAI), vol. 5459, pp. 394–401. Springer, Heidelberg (2009)
21. Kim, J.-T., Song, C.-S.: Comparison of Vocabulary Classification Systems among Thousand-Character Text, Yuhap, and Hunmongjahoi, Korean Literature Society, Linguistics and Literature, vol. 52, pp. 159–192 (1991) (written in Korean)
22. Jin, T.-H.: Problems in the Translations and Sounds of Thousand-Character Text. Hangeul-Chinese Character Culture 104, 80–82 (2008) (written in Korean)
23. Wikipedia: Section headers of a Chinese dictionary, http://en.wikipedia.org/wiki/Section_headers_of_a_Chinese_dictionary
24. Kingsoft2008 (谷歌金山词霸), http://g.iciba.com/
25. Classical Chinese Character FrequencyList, http://lingua.mtsu.edu/chinese-computing/statistics/char/list.php?Which=CL
26. YellowBridge, http://www.yellowbridge.com
27. CHINAKNOWLEDGE, http://www.chinaknowledge.de/Literature/radicals.html

Incremental Knowledge Acquisition Using Generalised RDR for Soccer Simulation

Angela Finlayson and Paul Compton

Department of Artificial Intelligence, School of Computer Science and Engineering,
The University of New South Wales, Sydney 2052, Australia

Abstract. This paper describes a system that allows soccer coaches to specify the behaviour of agents for the Robocup 2D soccer simulation domain [1]. The work we present is based on Generalised Ripple Down Rules [7,2] and allows the coach to interact directly with the system to incrementally model behaviours along with intermediate features during the knowledge acquisition process. The system was evaluated over a period of 6 months to measure the level of performance of the multi-agent teams created with the system and to gather feedback about the usability of the system. During this period the system was successfully used by four soccer coaches with differing levels of soccer and computer expertise. All coaches were able to use the system to develop teams that could play at a world class level against the finalists from the Robocup 2007 2D simulation tournament. The approach we present is general enough to be applied to any complex planning problem, with the requirement that a rich feature language is developed to support the specific domain.

1 Introduction

The Robocup 2d soccer server [1] is a useful simulation tool for developing and evaluating different approaches in the field of Artificial Intelligence, robotics and multi-agent systems. This domain has been of particular interest due to the underlying challenges of multi-agent co-ordination in a complex real-time environment with limited communication. With a search space that is prohibitively large, hand-coding of all possible situations and actions is impractical, and more general solutions are desired.

Early research in the soccer simulator domain focused on developing low level server synchronisation protocols, world model maintenance and low level skills such as kicking or intercepting the ball [3,4]. However, established teams such as CMU [17] , UvA [18] , Brainstormers [19] and HELIOS [20] have released code samples, providing solutions to many of these low-level problems. While the fine tuning of these lower level subsystems is still vital to team performance, the main challenge remains incorporating these low level skills into a coherent team strategy. Thus, most current research focuses more on higher level strategies such as planning and team co-operation and coordination [23,22,16].

Many researchers have focused on the adaptation of machine learning techniques such as reinforcement learning [21,32,33] to the domain. While there has

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 135–149, 2010.

been success adapting these methods to some of the sub-problems within the soccer domain, trying to learn goal-scoring behaviour from scratch with no innate expert knowledge seems to be very difficult for machine learning technique and other methods are needed to complement machine learning techniques. Although there has been a lot of research into the development of teams using domain independent techniques, there is still a great reliance on hand-coding of teams [29,30,31].

The drawback with machine learning or hand-coding is that often the tactics and strategies are buried within the code or the system. This can make modifications to strategies time-consuming, error-prone and difficult to debug [24]. Another problem is that specification of behaviours is limited to computer programmers rather than domain experts. Although domain experts can be consulted, much is lost in the translation. The soccer domain is an area where there is a vast amount of human knowledge about team strategies and coordination. It makes sense to tap into this knowledge and use it to allow agents to exhibit human like soccer playing abilities, especially when the goal of the Robocup community is "to beat the human soccer champions by the year 2050" [34]. Thus it would be desirable for domain experts(i.e. soccer coaches) to have a primary role in knowledge acquisition and to be able to directly interact with the system to produce teams and be able to add and refine knowledge to develop a team over time.

Teams such as the Dirty Dozen [24], KickOffTug [35] and the Headless Chickens [25] have attempted to address this issue. The Dirty Dozen developed SFLS, a rule based language that allows representation of team strategies in a rule base that can be modied easily by humans. However there is still the issue of added or modified rules interacting with existing rules which can result in undesired and inexplicable behaviours. This is the core maintenance problem of any rule-based system [14]. Another problem is that although they have focused on expressing plans and teamwork in easily modifiable human readable forms, specification of behaviours is still generally limited to computer programmers rather than domain experts. We are still left with the problem of how to get the relevant domain knowledge to put into the system.

KickOffTug [35] aimed to allow the decision making process to be designed by academics without the need of understanding the internal low level system. The system uses a graphical user interface that aids the creation of XML plans to define the agents behaviour and extract a decision regarding what action to perform next. In their system each plan is represented by a hierarchy of actions. Each action consists of a set of preconditions to check feasibility of the action, a set of post-conditions to check the success of the action, the low level action to be performed and a set of parameters for the action. The Headless Chickens focused on allowing domain experts using a graphical user interface to specify individual and team strategies. The interface allows the user to specify general strategies about player formations and passing and dribbling directions during different play modes. Styles could also be chosen such as whether a player has a preference for dribbling or passing the ball. These chosen styles determine priority levels

for the different kinds of behaviours during a game. Using this system, the user has limited control over the agent making it hard to predict agent behaviour. It is also difficult to determine reasons for unwanted behaviour [25].

In this paper we present an incremental knowledge acquisition framework that supports the creation of multi-agent teams in complex real-time environments and use our system to create teams for the Robocup soccer 2D Simulator. The underlying technique we use is a variant of the Ripple Down Rules (RDR) [6] technology and more specifically is a based on the generalised RDR approach [7,2]. Our approach aims to enable the process of eliciting knowledge of soccer strategies from a non-computer science soccer coach. The system provides an easy to use interface that allows an expert to monitor and refine soccer strategies in addition to incrementally building up their own language of higher level features and actions. The critical difference from other work in the area is the focus on gradual incremental refinement. Our motivation is to develop such techniques for complex interactive multi-agent environments with soccer as a challenging example. This system builds upon our initial prototype that was presented, along with preliminary results in [5].

The paper is organised as follows: The next section provides an overview of the knowledge acquisition methodology ripple-down rules(RDR) [6] giving examples of how it can be used by domain experts to easily create knowledge bases,and application areas that it has been applied to. We also discuss the actual RDR mechanism that we used in our system. In section 3 we discuss our framework for adapting RDR to the domain of robot soccer simulation and in Section 4, present our evaluation of the system. Finally in Section 5, we give our conclusions.

2 RDR Background

RDR is a tool that was developed to facilitate incremental knowledge acquisition by domain experts without the aid of a knowledge engineer. RDR was inspired by the observation that experts don't tend to give comprehensive explanations for their decision making. Rather they justify their conclusion given the context of the situation [14]. Based on this philosophy, certain features of RDR have emerged. The system gradually evolves over time while in use and validates any rules added to ensure that the addition of new knowledge does not degrade the previous knowledge base. Rules are added to the knowledge base to deal with specific cases where the system has made an error. These cases that prompted the addition of the new rule are stored along with the rule and are called cornerstone cases. The addition of new knowledge only requires the expert to identify features in a case that distinguish it from other cornerstone cases retrieved by the system and new knowledge is organised by the system, rather than the expert. These features provide the central difference between RDR and other knowledge acquisition approaches [7]. The advantage of this system is that the cumulative refinement over time allows the system to develop a high level of expertise, with the expert only expected to deal with individual errors. This approach contrasts with the intensive knowledge modelling approaches used in other knowledge acquisition systems.

2.1 RDR in Practice

RDR systems have been developed for a range of application areas and tasks. The first industrial demonstration of this approach was the PEIRS system, which provided clinical interpretations for pathology testing [28]. The approach has also been adapted and used for a number of tasks such as multiple classification [9], control [10], heuristic search [13], document management [15], configuration [11] and resource allocation [12]. There has also been work done in the area of combining machine learning techniques with RDR to reduce the amount of knowledge acquisition needed from an expert [8].

Studies comparing RDR to machine learning techniques have shown that RDR systems converge and end up with similar sized knowledge bases as those created by machine learning techniques [9,26] and that they cannot be compressed much by simple reorganisation [27]. However machine learning systems depend on well classified examples in sufficient numbers whereas an expert can provide a rule for a single case and a working system will start to evolve. Experts can also deal more successfully with single rare cases [8]. Some systems have integrated RDR with machine learning to allow a human to guide the machine learning process. One system combined a framework based upon NRDR with genetic algorithms [36]. This approach aimed to facilitate the formulation and tuning of operators involved in the development of the genetic algorithms. This was achieved by allowing a human to incrementally create rules to guide the process of offspring generation. Another system was built to allow experts to refine the output provided by machine learning in the area of object recognition [37]. In this system KA is combined with an incremental exception learner and the expert must decide whether any incorrect conclusions should be corrected at the machine learning level or the heuristic level. Further work followed on from this system to detect honey combing in lung images using a generalised RDR structure [38].

Overall these studies demonstrate that the application of RDR techniques can be used to provide simple and effective knowledge acquisition environments in a range of areas. There is now significant commercial experience of RDR confirming the efficiency of the approach. One company, Pacific Knowledge Systems supplies tools for pathologists to build systems to provide interpretative comments for medical Chemical Pathology reports. One of their customers now processes up to 13,000 patient reports per day through their RDR knowledge bases and have built about 10,000 rules, giving very highly patient-specific comments. They have a high level of satisfaction from their general practitioner clients and from the pathologists who keep on building more rules, or rather who keep on identifying distinguishing features to provide subtle and clinically valuable comments. A pathologist generally requires one or two days training and rule addition is a minor addition to their normal duties of checking reports; taking at most a few minutes per rule (Pacific Knowledge Systems, personal communication).

Although RDR has been comprehensively evaluated for classification problems and has been extended for use in other domains, it has never been explored in

the area of co-operation and planning in a multi-agent environment. Preliminary work was done by Kwok [39] investigating the use of RDR in a planning task. However, no results were actually published in this study. The blocks world domain used in the study differs from a multi-agent planning domain in that there are no time constraints, actions are deterministic and there is no hidden state. These characteristics allowed a STRIPS type planning approach to be used. However, this would not be suitable for a real-time, noisy multi-agent planning environment.

In this paper we present a system based on Generalised RDR [7,2] which supports the incremental modelling of intermediate features during the knowledge acquisition process, allowing experts to create their own abstractions of the domain. The inference mechanism used in our system is described in the next section.

2.2 RDR KB Structure and Inference Mechanism

The knowledge base structure and inference mechanism that we used in our system is based on the generalised RDR framework presented in [7,2]. In our system, a knowledge base is structured as a binary tree where every node can have a true (exception/correction) branch and an alternative or false (else-if/sibling) branch. Inference proceeds in a similar manner to that of SCRDR [40], however it is repeated over this structure producing an ordered list of output containing possibly multiple conclusions. Once a conclusion has been added to the output, the inference cycle potentially repeats, however any nodes that have already fired and added their conclusion to the output are not considered in the next cycle of inference. The knowledge base can also contain stopping conclusions. Stopping conclusions are added as exceptions and are used in a similar way to stopping rules in MCRDR [9]. This is explained further in the outline of the inference mechanism below:

1. Inference starts at the root node and when a rule fires a correction branch is taken, otherwise a sibling branch is taken. This process applies until there is no next node to evaluate. If there were no nodes that actually fired, the inference process has finished, otherwise a conclusion has been found. The new conclusion is located in the last fired node.
2. Further inference depends on the type of conclusion that was found. These can be stopping conclusions, intermediate conclusions or final conclusions.
   - Stopping conclusions are special conclusions that are not actually added to the output list when they fire, but cause the inference to continue with the next sibling node in the inference process. This has the effect of stopping the conclusion from the previously fired node, but allows inference to continue. Inference will either continue with the siblings of the stopping rule node if any exist or the siblings of the previously fired node otherwise. It is not necessary to repeat the inference cycle from the beginning of the tree when stopping conclusions are found, as no new input has been added to the output list. Thus, no new conclusions could possibly fire from the any of the preceding nodes in the tree.

- Intermediate conclusions are potentially added to the output list and can be used during subsequent inference to determine further conclusions. However, it is important that the output does not contain conflicting intermediate conclusions. Thus, once an intermediate conclusion has fired and is contained in the output list, no further conclusions from the same conclusion class can be added to the output list. For example if F1 = X is already contained in the output list then conclusion F1 = Y cannot be added to the output list. In this situation the conclusion is treated as a stopping conclusion and inference proceeds with the next sibling in the knowledge base. This results in a conflict resolution strategy that is handled by the chronological ordering of the sibling relationships in the tree. As soon as a conclusion from a new conclusion class is found and added to the output list, inference immediately repeats starting from the root node. Once a conclusion is added to the output list, it cannot be retracted.
- Final conclusions are conclusions that signify that no more knowledge is required and that no more inference is needed. This is important in domains where fast reaction times are needed. Once we have found a final conclusion we do not need any further inference and can act straight away.

This process is repeated from the root of the tree until no new conclusions can be added to the output list or until a final conclusion is reached. In general, the distinction between intermediate and final conclusions may not be necessary depending on the nature of the domain. In this case inference would end only when no further conclusions could be added to the output.

The inference mechanism differs slightly from the generalised RDR proposed in [7] and revised in [2], where evaluation starts from the beginning of the tree when stopping conclusions are found. However, the difference lies only in the efficiency of the algorithm, as the output would not be different under the proposed generalised RDR strategy. A more precise definition of the inference mechanism used in our system is presented in Algorithm 1.

Algorithm 1: Generalised RDR Inference

```
Evaluation(KB,case){
    Conclusion c; Output o;
    loop{
        c = repeatEval(root(KB),case,o);
        if (c is NOT NULL) addElement(o,c);
        if (c is NULL OR isFinalConclusion(c)) exit;
    }
}

Conclusion repeatEval(Node n, Case c, Output o){
    if(n is NULL)  return NULL;
```

```
    Conclusion c0 = conclusion(n);

    if (fires(n) AND NOT isUsed(o,n)){
       Conclusion correction = repeatEval(getCorrection(n),c,o);
       if(correction is NOT NULL AND NOT isStopping(correction)){
           return correction;
       }else if (NOT firedBefore(n)) {
           setUsed(n);
           if(isStop(c0)){
               Conclusion sibling = repeatEval(getSibling(n),c,o);
               if( sibling is NOT NULL) {
                   return sibling;
               }
           }
           if(NOT isContained(o,c0)){
               return c0;
           }
       }
   }
   return repeatEval(getSibling(n),c);
}
```

3 A Framework for Adapting RDR to the Soccer Simulation Domain

A system was developed to support the creation of Robocup 2D soccer agents that could incrementally acquire knowledge from human soccer coaches. This required implementing soccer agents that could be controlled by RDR KBs along with a graphical user interface to be used during the knowledge acquisition process. The collection of low level behaviours, the underlying world model and the synchronisation module for the RDR soccer agent were developed primarily using the CMU99 [17] and Helios [20] code bases.

3.1 RDR Soccer Agents

Soccer playing RDR agents that use RDR KBS to decide what actions to undertake at each simulation clock-cycle were created. Due to the requirements of the domain, each RDR agent corresponds to one individual soccer player and must make decisions independently based upon their own internal world models rather than having one centralised agent that controls every player in the team.

At each clock-cycle when the agent must decide what actions to perform, the RDR KBS maps the current case to possibly many user defined features and ultimately to particular actions. These actions are then executed by the agent. A case starts off consisting of all the primitive attributes from the world model as well as the user defined features and actions that were part of the RDR output

in the previous clock-cycle. The world model of the agent includes beliefs about the state of the game such as the location of the ball, the location of other players and the location of the agent itself. When the case is processed by the inference engine, any user defined features that fire are added to the case and can be used in the subsequent inference of that case. For example, if a feature called F1 was fired in the current cycle it would be added to the current case and rules that use F1 in their conditions could now possibly fire during the repeated inference process.

At each clock-cycle, each agent can choose only one primary action. This is determined by using the agent's primary knowledge base. Primary behaviours include behaviours that are responsible for movement of the agent's body, such as "Pass to player ahead" or "Intercept the ball". All agents with the exception of the goalie, share the one knowledge base that determines primary actions. Specialised behaviour for different positions can be achieved by the use of role attributes such as "Right Defender", "Sweeper", "Left Attacker" etc. The goal keeper has its own separate KB due to the specialised nature of the role. Along with a primary action, an agent may also choose to make a secondary action such as turning their head to look around the field. These actions were determined by a separate secondary KB. Since secondary actions, by nature often depend on the primary action, the output of the primary KB was supplied as input to the secondary KB. In our system, the secondary KB was shared by all players.

The rule language available to the expert consisted of 194 attributes and functions, including the primitive attributes from the agents world model as well as pre-defined higher level attributes and functions. The higher level attributes and functions provided information about concepts such as the closest teammate to the agent, the number of opponents in a given region of the field and the distance of the agent to particular points on the field. The rule language also included standard mathematical operators and functions. The expert could also choose to add to the rule language by defining their own user defined features. The data types available for both attributes and user defined features included simple types such as Integer, Float, Boolean, Nominal and Ordinal, along with more sophisticated types such as Player, Point, Region and Sets of players and points. A set of 22 primary actions were provided to be used as final conclusions in the KB. These included high level actions such as intercepting the ball, shooting, passing the ball to a particular player or dribbling to a particular point. Other actions were supplied that provided more low level control such as kicking the ball to a particular point on the field, moving to a particular point on the field, or turning to a particular point on the field. A total of 8 secondary actions was also supplied to control the direction of the agents head. These included actions to turn the head towards the ball, a particular player, a particular point on the field, or to simply scan the field.

The graphical user interface allows the expert to watch or step through a replay of a game and find incorrect behaviour. When a simulation is run, the attributes from each agents internal world models are logged to files. As the game is being rerun, the training window displays this information for each agent for

the current clock cycle. The Training Window also displays the RDR output for the particular player in the particular cycle from both the primary and secondary KBs. The output is an ordered list that contains any intermediate features, along with the final conclusions for the agent in that cycle.

When an undesirable behaviour is observed, the expert simply pauses the game replay, clicks on the appropriate agent and is then able to add new rules to the correct the behaviour. To do this the expert must determine the first incorrect conclusion in the output list and either choose an appropriate conclusion to replace it or simply choose to stop the conclusion. The user must create conditions to determine when the new rule will apply and once this is validated against previous corner stone cases, the rule is added as an exception in the KB. The output list is updated to show what the output would be with the modified KB. The expert continues this process until the output list contains no more incorrect conclusions. If the expert decides there should be additional conclusions in the output list, a conclusion may be added as a sibling in the KB. This process is repeated until the output list for the primary KB is correct. The expert can then optionally follow the same process for the secondary KB. This whole process can be repeated for different agents in different cycles. The expert then saves the updated KBs and new code for the agents' is automatically generated and compiled. At this stage one session of KA has been completed and the modified KBs are now used for making decisions for the agent in subsequent simulations.

4 Testing and Evaluation

Volunteer coaches were recruited with the aim of providing a mixture of skills and expertise for the KA evaluation process. A basic level of computer literacy and a basic level of interest and knowledge about soccer were the only prerequisites for participation. The first coach (KBRDRR) was the author and RDR researcher and had a minimal amount of real work soccer knowledge. The second coach (KBCOM) was a computer science lecturer with a high level of computer expertise who had an interest in watching soccer but had never played. The third volunteer (KBCOMSOC) was a computer science PhD student with a high level of computer expertise and a high level of soccer expertise, having played and coached soccer for many years. The final coach (KBSOC) had a high level of soccer expertise, having played and coached for many years, but had only basic computer literacy skills and no computer programming skills.

The volunteers were given a training session over a period of approximately 2 hours. Once the training session was over, the volunteers were able to have access to the built-in help facility of the system and were offered system support via email and face to face consultation. The volunteers were given a time frame of 6 months to build the KB for their team. In this time it was up to the volunteer as to how much or how little time was dedicated to the task. The coaches were given a set of 7 teams to use as competitors during the 6 month knowledge acquisition process. The KBs created by the four coaches were extensively evaluated against

a set of 11 teams that had not been used in the KA process and whose tactics and strategies were unknown. The performance of each KB was evaluated at different stages of the KB evolution to show the pattern of performance as rules were added. The teams chosen to be part of the evaluation or test set were teams from the 2007 Robocup Championships.

Experiments were run using the Robocup 2D Soccer Simulation Server Version 11.1.0. Each experiment was run on Pentium 4 2.8GHz machines with 1G RAM, with one machine dedicated to the soccer server and one machine dedicated to each team. Rule bases were evaluated at regular intervals to measure the performance of the KBs as rules were added. In our experiments, every rule base at every interval was evaluated by playing 20 games against each of the 11 test teams. An overall score percentage was determined based on adding up all the wins, multiplying by 3 and adding 1 for each draw. This is the method used in the Robocup 2007 championship to determine which teams make it to the final rounds of the tournament [41]. A selection of the results are shown in Table 1.

Table 1. Average Scores against a selection of the test teams for KBS from each coach. The first column indicates the rank of the test team in Robocup 2007. The number of rules in the KB at the time of evaluation is shown in brackets next to the score percentage and represents the best average performance of the KB over its evolution against the particular team.

	Team	KBRDRR	KBCOM	KBCOMSOC	KBSOC
1	Brainstormers	23% (58)	50% (280)	35.5% (40)	13% (20)
3	Helios	10% (112)	10% (520)	2% (20)	3% (20)
6	ATH	33% (100)	50% (747)	37% (60/260)	37% (104)
8	Nemesis	20% (112)	43% (660)	29% (40)	7% (20/104)
12	Rioone	67% (112)	97% (700)	93% (240)	33% (104)
13	YowAI	93% (112)	100% (240-747)	100% (40/60/141)	80% (104)
14	KickOffTug	100% (40-112)	100% (40-747)	100% (20-380)	80% (104)

Overall, the coaches were able to use the system effectively to express strategies and tactics and all coaches were able to produce teams that could play at a competitive level against the finalists from the Robocup 2007 2D simulation tournament. Due to the volunteer coaches having different amounts of time available to put into KA, the four final KBs were different in size. The final version of KBCOM contained 747 rules, KBCOMSOC 380 rules, KBRDRR 112 rules and KBSOC 104 rules. However the final rules bases did not necessarily result in the highest performances against different test teams. As can be seen from the results, all KBs achieved their best average performance against the brainstormers team at a relatively early point in their KB evolution. The addition of extra rules past that point did not increase the level of performance against that particular team. However performance against most of the other teams did increase with the addition of rules. For example, KBCOM performs best against ATH with its final KB of size 747 rules.

Another observation from our results is that performance against the different teams did not depend entirely on their rank from the RoboCup championships. For example all teams found Helios to be a much tougher competitor than Brainstormers, despite Brainstormers being ranked number 1. Similarly, all teams were able to perform at a higher level against ATH than Nemesis. However, it seemed clear that all teams were able to convincingly beat the lower ranked teams, such as YowAI and KickOffTug and were able to do so from quite early on in their rule base evolution. All teams except KBSOC were able to consistently win against Rioone, however it would have been interesting if the addition of more rules to KBSOC could have increased performance against this team.

During the KA process, the coaches expressed their frustrations by the difficulties imposed by the underlying soccer simulator. At times, the agents' world models contain noise, sometimes leading to incorrect decisions. The agents' actuators also contain noise, meaning that when an agent did make the right decision, the action did not always execute as intended. Other aspects of the soccer simulator placed limitations on the type of soccer strategies and tactics that could be used. For example, in many soccer books there is an emphasis on communication to aid teammwork, however the Robocup domain places severe restrictions on the communication that is allowed during a game. Another limitation in the 2D soccer server is the obvious lack of the third dimension. This had the most impact on the soccer coach with the most real life soccer skills, as many tactics envisaged involved the use of height. This particular coach found that this affected the strategies not only to do with the ball possessor, but the entire formations and positioning used throughout the game.

Another issue that arose was the fact that even though some of the experts had high levels of soccer knowledge, they did not necessarily have high levels of soccer simulation knowledge. This meant that during the beginning phases of KA they actually went through a learning period where they were trialling the actions and higher level attributes. This resulted in a large number of local patches that were results of the experts completely changing their minds, rather than refining the strategies used. Thus, large areas of the KB that were created in the initial stages became unused and obsolete. This invalid knowledge possibly hindered the efficient acquisition of new pieces of knowledge. The expert with the largest KB reported that there seemed to be a lot of repetition in the addition of the later rules, involving the addition of many similar local patches through the KB. Future work could involve methods to re-organise the KB to remove redundancy and improve efficiency, such as methods described in [27], or perhaps an adaptation of the work based on the minimum description length principle to delete obsolete knowledge [42].

The KBs performance did not increase monotonically as rules were added and KBs that performed well against one team, did not necessarily perform well against another team. The experts found that many of the tactics that worked against the easier teams did not necessarily work against the harder teams. Thus trying to improve performance against some of the tougher training teams involved total change of strategies which resulted in temporarily 'breaking' some

of the parts of the rule base. As the experts tuned their teams against the more difficult training teams it was noticed that the teams' performance against the easier teams tended to drop. Thus, the experts would then try to add rules to improve performance against the weaker teams. It proved to be difficult to produce one team that could optimally handle all opponents. This is possibly due to the nature of the domain, where trade-offs must be made between safe or defensive playing which will reduce the number of goals scored against the team and a more aggressive, attacking behaviour.

Despite the success of the teams created with the system, there were some opponents that none of the coaches were able to produce teams to beat. This is believed to be in part due to the underlying base code that was used, as no amount of strategy can create a champion team without the underlying low level code that it relies upon being highly tuned. In our research, we concentrated mainly on the knowledge level issues such as the rule language, the inference engine and the graphical user interface. Although we did provide access to some of the more low level aspects of the system, without a sophisticated underlying knowledge of physics and mathematics, it is not always feasible to expect a user to be able to by-pass the knowledge level abstractions provided. For example, a function was provided that calculates whether an agent is in a position to successfully shoot for a goal. The coach could either use the pre-defined attribute provided that may be sub-optimal or formulate their own rules for this kind of decision. However, this would require a level of mathematical ability that could not be expected from the average soccer coach. One of the soccer coaches who did have a strong background in maths and physics (KBCOMSOC), did attempt to write his own rules for various situations using complex mathematical formulae. However, the coach still had the desire to tune certain parameters in these formulae, and ended up breaking the golden RDR KA rule and editing the KB manually. This coach going through a process of trial and error to try to find the right combination of parameters. Due to the noise in the system, this proved to be a futile task. It should be noted that this KB that had been manually edited showed the most unstable behaviour out of all the KBs.

One of the reviewers suggested that meta-knowledge about strategies against different teams may be useful. This is certainly worth exploring, but we expect that the real underlying problem is an expert's ability to construct theories/models/justifications about a domain given the language available. RDR with its use of case differences, minimises the need for the expert to understand the whole model of the domain being created, but the need remains to be able to find insights into how to play better and then articulate these insights with the language provided. This study highlights this problem.

5 Conclusion

In general our conclusion is that RDR provides an effective way of dealing with multi-agent planning problems. The approach we present is general enough to be applied to any complex planning problem, with the requirement that a rich

feature language is developed to support the specific domain. Our studies confirm the intuitive conclusion that extra knowledge will not necessarily suffice if the underlying set of features is not powerful enough. Secondly and related: although GRDR seems to be a useful framework for allowing experts to create their own layers of abstractions, some abstractions need to be expressed in low level code below the knowledge level. Thirdly: although the GRDR is a powerful approach to addressing a very wide range of problems, it needs further integration with tools to restructure the KB. It seems likely, as we experienced here, that in more complex domains, expertise is likely to be much more experimental so that a more radical reorganisation of the KB that can be provided by local refinement, may be appropriate.

Acknowledgements

We would like to thank Victor Jauregui and Rex Kwok for their invaluable contributions to this research.

References

1. Noda, I., Matsubara, H., Hiraki, K., Frank, I.: Soccer server: A tool for research on mulitagent systems. Applied Artificial Intelligence 12, 233–250 (1998)
2. Compton, P., Cao, T., Kerr, J.: Generalising Incremental Knowledge Acquisition. In: Proceedings of the Pacific Knowledge Acquisition Workshop (2004)
3. Stone, P.: Layered Learning in Multi-agent Systems. PhD Thesis, Carnegie Mellon University (1998)
4. de Boer, R., Kok, J.: The Incremental Development of a Synthetic Mutli-Agent System: The UvA Trilean 2001 Robotic Soccer Simulation Team. Master's thesis, University of Amsterdam (2002)
5. Finlayson, A., Compton, P.: Incremental Knowledge Acquisition using RDR for Soccer Simulation. In: Proceedings of the Pacific Knowledge Acquisition Workshop (2004)
6. Preston, P., Edwards, E., Compton, P.: A 2000 Rule Expert System Without Knowledge Engineers. In: Proceedings of the 8th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada, pp. 17.1–17.10 (1994)
7. Compton, P., Richards, D.: Generalising Ripple-Down Rules. In: Dieng, R., Corby, O. (eds.) EKAW 2000. LNCS (LNAI), vol. 1937, pp. 380–386. Springer, Heidelberg (2000)
8. Suryanto, H., Compton, P.: Invented Knowledge Predicates to Reduce Knowledge Acquisition Effort. In: Tecuci, G., Aha, D., Boicu, M., Cox, M., Ferguson, G. (eds.) Proceedings of the IJCAI 2003 Workshop on Mixed-Initiative Intelligent Systems, Eighteenth International Joint Conference on Artificial Intelligence, Austin tate, Acapulco Mexico, August 9-15, pp. 107–114 (2003)
9. Kang, B., Compton, P., Preston, P.: Multiple Classification Ripple Down Rules: Evaluation and Possibilities. In: Proceedings of the 9th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada, pp. 17.1–17.20. University of Calgary (1995)

10. Shiraz, G., Sammut, C.: Acquiring Control Knowledge from Examples Using Ripple-down Rules and Machine Learning. In: Gaines, B.R., Musen, M. (eds.) Proceedings of Eleventh Workshop on Knowledge Acquisition, Modelgin and Management (KAW 1998), Banff, Alberta Canada, April 18-23, pp. KAT-5.1–KAT-5.17. University of Calgary, Calgary (1990)
11. Compton, P., Ramadan, Z., Preston, P., Le-Gia, T., Chellen, V., Mullholland, M.: A trade-off between domain knowledge and problem-solving method power. In: Gaines, B., Musen, M. (eds.) 11th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, SHARE, Banff, vol. 17, pp. 1–19. SRDG Publications/University of Calgary (1999)
12. Richards, D., Compton, P.: Revisiting Sisyphus I - an Incremental Approach to Resource Allocation Using Ripple-Down Rules. In: Gaines, B., Kremer, R., Musen, M. (eds.) 12th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, pp. 7-7.1–7.20. SRDG Publications/University of Calgary (1999)
13. Beydoun, G., Hoffman, A.: Incremental Acquisition of Search Knowledge. International Journal of Human Computer Studies 52(3), 493–530 (2000)
14. Compton, P., Horn, R., Quinlan, R., Lazarus, L.: Maintaining an expert system in J. R. Quinlan. In: Applications of Expert Systems, pp. 366–385. Addison Wesley, London (1989)
15. Kang, B., Yoshida, K., Motoda, H., Compton, P.: A help desk system with Intelligent Interface. Applied Artificial Intelligence 11(7-8), 611–631 (1997)
16. Obst, O., Bodecker, J.: Flexible Coordination of Multiagent Team Behaviour Using HTN Planning. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, pp. 521–528. Springer, Heidelberg (2006)
17. CMU RoboCup Simulator Team Homepage: Carnegie Mellon University, Pittsburgh, PA (2003), http://www-2.cs.cmu.edu/~pstone/RoboCup/CMUnited-sim.html
18. The Official UvA Trilearn Website: University of Amsterdam (2001), http://www.science.uva.nl/~jellekok/robocup
19. Brainstormers base source code, http://www.ni.uos.de
20. Helios base source code, http://sourceforge.jp/projects/rctools
21. Riedmiller, M., Merke, A.: Using Machine Learning Techniques in Complex Multi-Agent Domains. In: Stamatescu, I., Menzel, W., Richter, M., Ratsch, U. (eds.) Perspectives on Adaptivity and Learning. LNCS. Springer, Heidelberg (2002)
22. Kok, J., Spaan, M., Vlassis, N.: Multi-robot decision making using coordination graphs. In: de Almeida, A.T., Nunes, U. (eds.) Proceedings of the 11th International Conference on Advanced Robotics, ICAR 2003, Coimbra, Portugal, June 30-July 3, pp. 1124–1129 (2003)
23. Yunpeng, C., Jiang, C., Jinyi, Y., Li, S.: Global Planning from Local Eyeshot: An Implementation of Observation -based Plan Coordination in RoboCup Simulation Games. In: Birk, A., Coradeschi, S., Tadokoro, S. (eds.) RoboCup 2001. LNCS (LNAI), vol. 2377, p. 12. Springer, Heidelberg (2002)
24. Buttinger, S., Diedrich, M., Hennig, L., Honemann, A., Hugelmeyer, P., Nie, A., Pegam, A., Rogowski, C., Rollinger, C., Steffens, T., Teiken, W.: The ORCA Project Report, http://www.cl-ki.uni-osnabrueck.de/~tsteffen/orcapub.html
25. Scerri, P., Coradeschi, S., Torne, A.: A user oriented system for developing behaviour based agents. In: Asada, M., Kitano, H. (eds.) RoboCup 1998. LNCS (LNAI), vol. 1604, pp. 173–186. Springer, Heidelberg (1999)

26. Compton, P., Preston, P., Kang, B., Yip, T.: Local patching produces compact knowledge bases. In: Steels, L., Van de Velde, W., Schreiber, G. (eds.) EKAW 1994. LNCS, vol. 867, pp. 103–117. Springer, Heidelberg (1994)
27. Suryanto, H., Richards, D., Compton, P.: The Automatic Compression of Multiple Classification Ripple Down Rule Knowledge Base Systems: Preliminary Experiments. In: Proceedings of the Third International Conference on Knowledge-Based Intelligent Information Engineering Systems, L. Jain Adelaide, pp. 203–206 (1999)
28. Edwards, G., Compton, P., Malor, R., Srinivasan, A., Lazarus, L.: PEIRS: a pathologist maintained expert system for the interpretation of chemical pathology reports. Pathology 25, 27–34 (1993)
29. Asakawa, H., Ueda, M., Yamazaki, Y., Takeuchi, I.: YowAI 2007 Team Description. In: RoboCup 2007. LNCS (LNAI), vol. 2377. Springer, Heidelberg (2007)
30. Berger, R., Burkhard, H.: AT Humboldt Team Description 2007. In: Robocup 2007. LNCS (LNAI), vol. 2377. Springer, Heidelberg (2007)
31. Norouzitallab, M.: Nemesis 2D - Team Description 2007. In: Robocup 2007. LNCS (LNAI), vol. 2377. Springer, Heidelberg (2007)
32. Akiyama, H.: HELIOS 2007 Team Description. In: RoboCup 2007. LNCS (LNAI), vol. 2377. Springer, Heidelberg (2007)
33. Riedmiller, M., Gabel, T.: Brainstormers 2D Team Description 2007. In: RoboCup 2007. LNCS (LNAI), vol. 2377. Springer, Heidelberg (2007)
34. Kitano, H., Kuniyoshi, M., Noda, Y., Osawa, E.: RoboCup: The Robot World Cup Initiative. In: The First International Conference on Autonomous Agents (1997)
35. Gspandl, S., Monichi, D., Reip, M., Steinbauer, G., Wolfram, M., Zehentnerm, C.: KickOffTug - Team Description Paper 2007. In: Robocup 2007. LNCS (LNAI), vol. 2377. Springer, Heidelberg (2007)
36. Bekmann, J., Hoffman, A.: Improved Knowledge Acquisition for High Performance Heuristic Search. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 41–46 (2005)
37. Kerr, J., Compton, P.: Toward Generic Model-Based Object Recognition by Knowledge Acquisition and Machine Learning. In: Workshop on Mixed-Initiative Intelligent Systems, Int. Joint Conf. on AI (2003)
38. Singh, P., Compton, P.: Combining machine learning and heuristic rules using GRDR for detection of honeycombing in HRCT lung images. In: 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems,
39. Kwok, R.: Using Ripple Down Rules for Actions and Planning. In: Proceedings of the 7th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence (2002)
40. Compton, P., Jansen, B.: A philosophical basis for knowledge acquisition. Knowledge Acquisition 2, 241–257 (1990)
41. Robocup 2007 Soccer Simulation Competition (2007), http://wiki.cc.gatech.edu/robocup/index.php/Soccer_Simulation
42. Yoshida, T., Wada, T., Motoda, H., Washio, T.: Adaptive Ripple Down Rules method based on minimum description length principle. Intelligent Data Analysis 8(3) (2004)

Incremental System Engineering Using Process Networks

Avishkar Misra, Arcot Sowmya, and Paul Compton

School of Computer Science & Engineering, University of New South Wales
Sydney, Australia
{amisra,sowmya,compton}@cse.unsw.edu.au

Abstract. Engineering of complex intelligent systems often requires experts to decompose the task into smaller constituent processes. This allows the domain experts to identify and solve specific sub-tasks, which collectively solve the system's goals. The engineering of individual processes and their relationships represent a knowledge acquisition challenge, which is complicated by incremental ad-hoc revisions that are inevitable in light of evolving data and expertise. Incremental revisions introduce a risk of degrading the system and limit experts' ability to build complex intelligent systems. We present an incremental engineering method called ProcessNet that structures incremental ad-hoc changes to a system and mitigates the risks of the changes degrading the system. A medical image analysis application developed using ProcessNet demonstrates that despite a large number of ad-hoc, incremental changes the system's ability and accuracy in segmenting multiple anatomical regions in High Resolution Computed Tomography (HRCT) scans continue to improve.

Keywords: Knowledge Acquisition, Incremental Software Engineering, Computer Vision, Medical Image Analysis.

1 Introduction

Knowledge is commonly accepted as a fundamental component of any intelligent system. Such systems require knowledge to describe not just the domain specific attributes for a given problem, but also the control knowledge pertaining to how the algorithms should be combined intelligently to solve the problem. The challenge in constructing any intelligent system often lies in acquiring the knowledge from either labeled data via inductive pattern recognition and machine learning techniques [1], or directly from experts via knowledge engineering methods [2].

While these are commonly accepted approaches to acquire the knowledge, what is seldom considered is how the systems using this knowledge should evolve over time with the evolving data and level of expertise. The distribution of data available may change over time, bringing to light previously unseen cases. Alternatively new techniques or algorithms may be discovered as the expertise in the domain and problem evolves. As a consequence the underlying domain and control knowledge for a given system must also change over time.

The challenges of incrementally acquiring knowledge in the form of rules from experts have been successfully addressed by Ripple Down Rules (RDR) [3]. RDR's

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 150–164, 2010.

incremental revision strategy validates each change to a rule against existing cornerstone cases to ensure that the changes do not degrade the knowledge within the knowledge base. RDRs have been successfully employed to solve a number of challenging tasks in a variety of domains including [4 - 7]. RDR however only manage the knowledge within the rules of a knowledge base and ignore the knowledge within the rule interpreters, the library functions or source code of various algorithms that act on RDR's concluding inference. Also RDR capture and validate knowledge for a specific task in isolation.

As the complexity of a system grows however, it is often required to break it up into multiple parts rather than treat it as a single monolithic system. By dividing a system into smaller processes, we can identify key tasks and their influence on each other as they collectively solve the system's goals. This not only makes the task of engineering complex systems manageable, but also allows multiple experts to collaborate on constructing the system by focusing on processes within their area of expertise.

The knowledge acquisition for such intelligent systems must thus extend beyond just the acquisition of knowledge at a single process and a single expert. It must also consider the knowledge of how one process interacts with others. In a system of processes, modification at a process may change its output, which would in turn impact upon other processes that use that output. The dependencies between processes thus introduce a risk that a change may adversely degrade the performance of other processes and in turn the entire system.

In this work we propose a method called ProcessNet to incrementally engineer a network of processes in light of improving data, and evolving expertise and techniques, whilst managing the risks of ad-hoc changes degrading the system. ProcessNet addresses the knowledge acquisition bottlenecks faced when engineering all aspects of tacit and explicit knowledge for multiple processes, allowing large intelligent systems to be developed and adapted over time.

We have used ProcessNet to incrementally engineer a complex medical image analysis system, which automatically segments (i.e. delineates) multiple anatomical regions in High Resolution Computed Tomography (HRCT) scans of the lungs. The system, starting with no knowledge, is incrementally engineered to a complex network of 25 processes, each using a variety of computer vision algorithms, rules systems or pattern recognition techniques. In this study we present the system's improving accuracy in segmenting lungs, spine, sternum and shoulder as evaluated against hand-marked ground truth of 342 images from 20 patient studies. We also present the system's ability to accurately segment lung regions on a larger set of 583 images from 40 patient studies. The results show that despite more than 221 changes to different parts of the system, the system's processes continued to improve as better training data and improved vision expertise become available.

Engineering of software systems via such a generalized RDR-based knowledge acquisition technique has been previously suggested but not realized [8]. Other works such as [9] have been broadly framed within the generalized RDR approach. ProcessNet is the first development of a system where dealing with many components in an incremental fashion was essential. It was motivated by generalized RDR but has developed a specific approach to deal with the complex domain of computer

vision. ProcessNet provides a concrete methodology and demonstrates its ability to incrementally engineer a medical image segmentation system that simultaneously segments multiple anatomical regions.

2 ProcessNet

We propose a strategy called ProcessNet that minimizes the risk of degrading a system when engineered incrementally. ProcessNet extends the ideas of incremental validated revisions in Ripple Down Rules (RDR) [3] to the engineering of software systems. RDR are an incremental knowledge acquisition technique that encode knowledge in a nested hierarchy of rules and their exceptions. An expert can quickly correct errors in an RDR knowledge base by adding an exception rule that overrides the failing rule.

The fundamental idea of RDR is that any change to the knowledge must only be allowed if it is consistent with the existing cases that led to the knowledge thus far. To ensure this, each rule within an RDR knowledge base is supported by cases that prompted the addition or modification of that rule, known as *cornerstone cases*. Adding a rule or modifying an existing rule requires each of the potentially affected cornerstones to be checked so as to ensure that they continue to be handled correctly. This validated change strategy is a reasonable and effective approach to structuring what is typically ad-hoc knowledge acquisition in incremental settings and has been successfully applied to many areas including automatic interpretation of pathology results [4] and VLSI chip design [5].

In ProcessRDR [6, 7], multiple RDRs were used to incrementally acquire domain and control knowledge to guide image-processing tasks in a computer vision system. Each RDR knowledge base captures knowledge on algorithm selection, parameter tuning of algorithms and classification for a specific image analysis task. A change to rules within the knowledge base is validated using the rule's cornerstones, just as in RDR. However, changes made to source code and algorithms outside the rules are not managed, nor the impact of changes across multiple interdependent processes. Consider a process A within a system that measures features from an image and provides those features for subsequent inference by another process B. Changes in A can change the nature of the features produced by A. These changes may be significant enough to cause the values of features to lie beyond the range of values expected by process B and consequently affect B's ability to process new and even existing cases correctly. Validation to evaluate the impact of changes across processes becomes critical as the number of processes within a system grows. ProcessNet addresses these limitations.

ProcessNet considers a system as a *network of processes*, each responsible for a task, and that collectively solve the overall system goals. The system can be represented as a directed graph, where each process is a node and a directed edge between them represents the information flow between processes. Note that we only consider a directed acyclic graph with no circular dependencies. A sorted topology of the system's dependency graph gives us the order in which the processes must be executed, such that data requirements for each process are satisfied.

We define a *process* in very general terms to encapsulate arbitrary degrees of complexity and forms of knowledge expression such as source code, rule systems or machine-learned concept functions. The core functions of (a) a typical knowledge-based process and (b) a fixed algorithm process are shown in Fig 1. A process' knowledge may be impacted by changes to:

(i) data from other processes, which serve as input to the process (raw input).
(ii) derivation and expression of features, which are used by an inference system (feature extraction).
(iii) inference system that determines the algorithm or parameters to use (control knowledge).
(iv) the actual processing algorithms, parameter mappings and labels, which are used as actions of the process (algorithms and parameters).

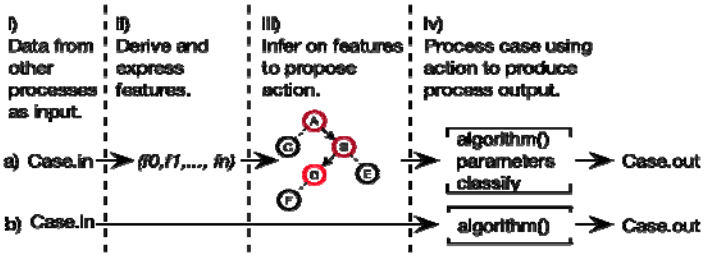


Fig. 1. A process' internals: (a) Knowledge-based process (b) fixed algorithm process

These four types of changes may arise due to a number of different factors. In a network of processes:

(i) Changes in one process' output may naturally impact another process' input.
(ii) Experts often invent new features or project these features into new feature spaces leading to changes in the derivation and expression of features.
(iii) The inference system may be constructed using machine learning techniques and may predict different actions in response to different features. Improvements to the inference system may occur because more labeled data becomes available for training, or revisions are made to the learning algorithm and its parameters.
(iv) The algorithms executed based on the actions proposed by the inference system, may be improved independently of the inference system.

A process representing a fixed algorithm may face changes to its input (i) and the algorithm itself (iv), whilst a knowledge-based vision process may encounter changes to any of the four functions as shown in Fig 1. The previously proposed solutions in ProcessRDR [6] only monitored and validated changes to the inference system (iii), and did not present a strategy to manage the impact of changes across multiple inter-dependent processes, which becomes critical as the number of processes within a system grows.

In keeping with RDR, we define a *case* at each process in terms of the process' input data and the resulting output. This ensures that a process only considers the minimal set of data sources necessary to undertake its task and captures process-specific

behavioral requirements. Like RDR, we require that any change made to a process be motivated by a case and captured as a cornerstone case for the process.

If a process is modified we evaluate the quality of the change by evaluating its cornerstone cases. Any difference to the input and output data for a cornerstone case highlights a potential effect as a consequence of these changes. We call these differences *cornerstone shifts*. The key difference between RDR and ProcessNet is in the handling of cornerstone shifts.

As is expected with evolving insight into the domain, our notion of truth and the nature of data may change. To accommodate for this, ProcessNet must allow for shifts in cornerstones, as long as they are consistent with our present notion of truth and deemed to be acceptable by the expert. This is in contrast to RDR where deviations in cornerstones are rejected outright. Permitting cornerstone shifts ensures that our knowledge continues to evolve and keep pace with 'concept drift'. Cornerstone shifts that conflict with ground truth are not allowed and will require vision experts to revise the changes, or make further changes to ensure consistent operation of the system.

Cornerstones for a process capture an example of the requirements for that given process, but may not capture the implicit contract of expectation that other dependent processes have with the process. So while we validate a process' correctness against its own cornerstones, the validation of its impact on dependent processes lies with the latter. Each of the dependent processes must also be evaluated for changes to the input and subsequent output, by checking against its own set of cornerstones. This frees the expert from having to consider the entire system when revising a single process. An expert improves the correctness of a specific process with respect to its cornerstones, with the expectation that other processes will also be revised in turn to accommodate for the changes as required.

Detecting cornerstone shifts and revising the processes to handle these shifts may continue recursively down the dependency graph of processes. The order in which the processes are evaluated and revised is determined by the order in which they are run within the system to satisfy the data requirements. The incremental revision and validation of individual processes, and collectively the entire system, should ensure that the system continues to adapt in light of more labeled data and evolving expertise.

3 Application to Medical Image Segmentation

3.1 Medical Image Segmentation

Medical imaging has become an invaluable resource for physicians across a number of modalities and applications, by providing a non-invasive look inside a patient to detect, diagnose and monitor diseases. A computer aided detection and diagnosis system (CAD) [10] could not only support physicians in diagnosis, but also assist surgical interventions. In order to support such applications, medical image analysis systems capable of automatically detecting and delineating or *segmenting* anatomy become essential. The segmented anatomy can then be used by CAD systems to detect abnormalities indicative of diseases or to develop patient specific 3D models for visualization that support surgical intervention.

The development of anatomy segmentation systems, however, is a challenging task due to the complexity of the domain, natural variations between patients, impact of disease on normal anatomy and even variations between experts on delineation of ground truth. These systems require domain knowledge about the anatomy and the imaging modality, as well as control knowledge on interpreting and guiding the image processing necessary to segment and label the results [11]. The complexity of the domains and deformation of the natural anatomy due to the presence of diseases also means that secondary regions must be segmented and used as cues to direct segmentation of other anatomical structures. This strategy of segmenting by parts is often represented as a network of processes or as agents [12]. Since medical image segmentation systems are a specific application of computer vision systems, we look at some of the general approaches to the engineering of computer vision systems, and the incremental knowledge acquisition bottlenecks that they face.

Expert-based vision systems [2, 13, 14] rely on vision experts to explicitly encode the knowledge in the form of algorithms or rules within a process and the relationship between processes within a network. These techniques suffer from two main limitations. Firstly, experts find it difficult to articulate all knowledge completely and accurately. So revisions are often required to correct knowledge previously declared. Secondly, once captured, this knowledge is difficult to update and revise. The vision experts are expected to add to this knowledge and revise imprecise knowledge without adversely affecting other parts of the system that depend on it. This becomes more challenging as the system's complexity grows, and increases the number of system components potentially affected. These issues are the same as those identified earlier to affect most intelligent systems, commonly referred to as knowledge acquisition bottlenecks.

More recently, vision researchers have sought to formulate vision systems in mathematically sound frameworks [15]. In essence these approaches attempt to select a sequence of processing steps that reduce the overall cost of making a poor segmentation. Each processing step represents a computer vision algorithm and is selected from a set of available algorithms, based on a reward or cost function. This function is either explicitly encoded by a vision expert or induced from. Pattern recognition and machine learning techniques [1] are used [16, 17] to approximate the reward or cost function from a set of labelled training data. In the context of medical imaging, this would require medical experts to mark accurate ground truth for a large number of training images, which can be difficult due to the demands on the expert's time.

Graph Cut [18 - 20] based segmentation techniques are an example of an expert-defined cost function for inference at an image pixel level. Even though these techniques avoid the requirement of training data, an expert must still define the cost function and its parameters manually, often via trial and error. In a complex system with multiple processes, this presents a daunting challenge for vision experts. The knowledge acquisition bottleneck has shifted to experts requiring a strong understanding of the even more specialised task of engineering cost functions.

None of the approaches have actually addressed the core issue of the knowledge acquisition bottleneck. Even though some are based on sound mathematical foundations, the underlying features or algorithms are still heuristically composed by experts and revised incrementally in response to newly available data as well as the expert's evolving understanding of the problem and vision algorithms. Engineering of medical anatomy segmentation systems are built by incremental ad-hoc engineering and face

all the issues identified earlier in Section 1. This makes the medical image segmentation problem not only an important one to address, but also well suited for incremental engineering via ProcessNet.

3.2 Lung Anatomy Segmentation

We have put the ideas of ProcessNet into practice and developed a lung anatomy segmentation system that segments parts of the thoracic anatomy visible within High Resolution Computed Tomography (HRCT) studies of patients. While lung anatomy segmentation systems have been developed for a number of modalities including x-rays [21] and HRCT [22], none of the approaches has attempted to segment multiple anatomical regions in parallel, nor have they addressed the risks inherent in incremental ad-hoc revisions.

Our lung anatomy segmentation system segments the spine, sternum, shoulder blades, trachea, bronchi, oesophagus and lungs. The system is constructed to process sparse HRCT studies with an average of 15 images of 1 mm thickness, spaced every 15 mm along the axial plane of the body. The HRCT data is from real patients suffering from a variety of diffuse lung diseases that affect the lung regions.

The anatomy segmentation system starts with no prior knowledge about the domain or any vision algorithms. A vision expert adds knowledge incrementally as required in the form of processes and their relationships. During development, the expert seeks to ensure that both the ability of the system to undertake additional segmentation tasks and the accuracy with which a task is carried out are improved. Segmenting multiple regions in parallel allows processes of the system to take advantage of anatomical cues and partial registrations, for subsequent segmentations of other anatomy by other processes. The vision expert can define a process in source code that calls upon a library of fundamental image processing functions (for example [23]) or implements specific vision algorithms within the process itself.

3.3 Results

ProcessNet's incremental validated change strategy ensures that the performance of the lung anatomy segmentation system is either improved or maintained consistently, despite frequent revisions by an expert as they attempt to improve parts of the system. To establish this, we measure the system's performance over three training phases – A, B and C. During each training phase, a vision expert engineers the anatomy segmentation system using 3 patient studies each. Three new studies are added at each phase and the system is adapted to deal with them. At the end of each training phase, the system segmentation accuracy is evaluated against independent test sets that were not used in training. Details of the incremental software engineering of the system using ProcessNet and the evaluation of segmentation accuracy over the three training phases are presented below.

3.3.1 ProcessNet in Operation

In the training phases, a process is added or modified as the need arises for better features or improved techniques to handle a specific task. Alternatively processes that perform poorly or become redundant are removed from the system. A summary of revisions during the training phase A, B and C is presented in Table 1.

Table 1. Revision summary for training phases A, B and C.

	A	B	C	Total
Studies	3	3	3	9
Images	59	47	58	164
System Revisions	47	6	8	61
- Process Additions	22	4	3	30
- Process Modifications	147	51	23	221
- Process Removal	0	1	3	4
Number of Processes	22	25	25	25
Cornerstones Added	51	45	38	134

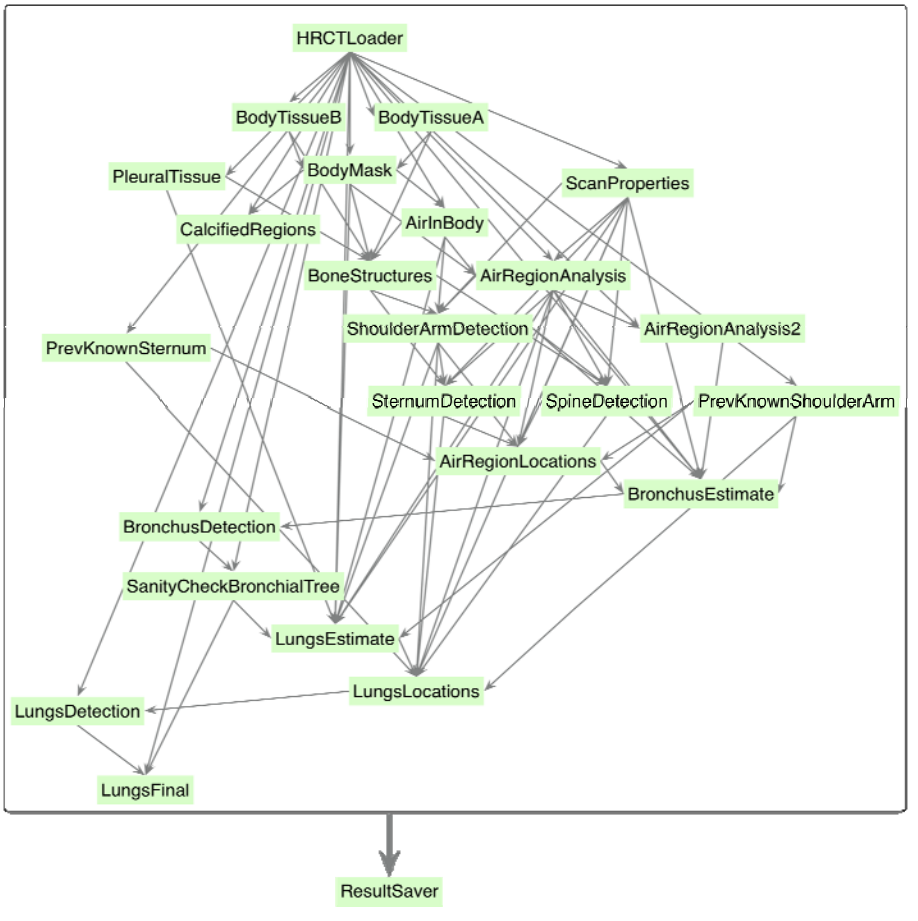


Fig. 2. Lung anatomy segmentation system after Phase C. Arrows indicate information flow between the processes. ResultSaver is a process that receives input from every other process.

Table 2. Processes listed in the run order

ID	Process	Purpose	Requires
1.	HRCTLoader	Loads and enhances DICOM images.	-
2.	PrevKnownShoulderArm	Loads previously segmented shoulder regions.	1
3.	PleuralTissue	Segments pleural tissue.	1
4.	BodyTissueB	Segments fatty body tissue.	1
5.	BodyTissueA	Segments muscular body tissue.	1
6.	BodyMask	Generates a mask of the body.	1, 4, 5
7.	AirInBody	Detects air filled regions within the body.	1, 6
8.	BoneStructures	Detects bone structures.	3-7
9.	CalcifiedRegions	Detects calcified regions.	1, 6
10.	PrevKnownSternum	Loads previously segmented sternum regions.	1
11.	ScanProperties	Extract scan information from DICOM header.	1
12.	AirRegionAnalysis	Measures structural features for air regions.	1, 6, 7, 11
13.	ShoulderArmDetection	Detects shoulder, arms and shoulder blades.	7, 8, 11, 12
14.	SpineDetection	Detects the bone structures of the spine.	8, 11-13
15.	SternumDetection	Detects the bone structures of the sternum.	8, 11, 12, 13
16.	AirRegionLocations	Measures location based features for air regions.	1, 12
17.	AirRegionAnalysis2	Measures textural features for air regions.	2, 10-15
18.	BronchusEstimate	Estimates bronchus, trachea and oesophagus.	1, 2, 11-13, 16, 17
19.	BronchusDetection	Labels bronchus, trachea and oesophgaus.	1, 18
20.	SanityCheckBronchialTree	Resolve conflicts within bronchus regions.	1, 19
21.	LungsEstimate	Estimates regions likely to be lung.	1-3, 6, 7, 11-13, 20
22.	LungsLocation	Measures location based features for lungs.	2, 10-15, 21
23.	LungsDetection	Labels lungs as left, right or merged.	1, 22
24.	LungsFinal	Resolves conflicts between lung and bronchi.	1, 20, 23
25.	ResultSaver	Saves results of processes as XML and images.	1-24

During training phase A, there were 47 system revisions or training events, where a vision expert revised parts of the system; these diminished to 6 and 8 by phases B and C respectively. Each training event involved changes to one or more processes as new processes were added and existing ones modified or removed.

In phase A, 22 new processes were added and subsequently modified 147 times during the course of training as issues were discovered in the light of new training data. The vision algorithms necessary for the system were being introduced and subsequently refined during this period. The processes implemented basic image-processing algorithms such as image filtering, thresholding, morphology, image subtraction and region growing. By training phases B and C, the number of new processes added had reduced to 4 and 3 respectively and the modifications made to processes also decreased to 51 and 23 respectively.

In phases B and C, 1 and 3 processes respectively were removed. This is because some processes were found not to perform well. The new processes added in lieu of these processes offered better algorithms to carry out specific tasks and the over-all system goals. For example the lungs detection processes evolved from a simple thresholding algorithm in phase A, to applying locally parameterized active contours in phase B and finally superseded by a statistical model and an RDR classifier using relational features defined with respect to spine, sternum, left and right shoulders in phase C. This is driven by the expert's choice, but ProcessNet supports the decisions by providing for consistency checks on cornerstones.

At the end of phase C, the 164 images available for training had been used to add 134 cornerstones to 25 processes within the system. Further analysis of the cornerstones showed that these 134 cornerstones came from 97 unique training images. Since each process maintains its own set of cornerstone cases, a new training image that results in revisions to multiple processes may lead to multiple cornerstones, each derived for the process being revised.

In total the vision expert made 221 revisions to processes in the form of source code edits, addition of new rules to knowledge bases or invocation of machine learning algorithms for induction of rules at a process. Further analysis reveals more than 70% of the modifications were source code edits, which would not have been validated under ProcessRDR. The final network of processes after phase C is shown in Fig. 2, and details about the processes and their dependencies are shown in Table 2.

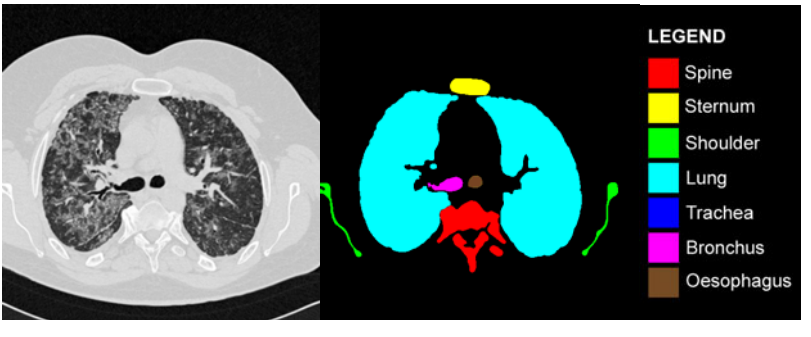


Fig. 3. Anatomy segmentation: (*Left*) Original image, (*Right*) Segmentation results

3.3.2 Evaluation of Segmentation Accuracy

At the end of each training phase, the accuracy of the system in segmenting anatomy was evaluated on two independent test sets of hand-marked ground-truth. The first test set, called Anatomy-20, contains ground truth image masks for lungs, spine, sternum and shoulder regions for each of the 342 images from 20 patient studies. The second test set, called Lungs-40, contains ground truth masks for only lung regions in 583 images from 40 patient studies. The 20 patient studies in the Anatomy-20 test set is a subset of the 40 patient studies in the Lungs-40 test set.

The segmentation results of the system are compared against the hand-marked ground truth masks for spine, sternum, shoulder and lung regions using the metrics of sensitivity and specificity defined below:

$$\text{Sensitivity} = TP / (TP+FN) .$$

$$\text{Specificity} = TN / (TN+FP) .$$

where,

TP = True Positives are the number of pixels correctly labeled as lungs.
TN = True Negatives are the number of pixels correctly labeled as non-lung.
FP = False Positives are the number of pixels incorrectly labeled as lungs.
FN = False Negatives are the number of pixels incorrectly labeled as non-lung.

A good segmentation system should have high sensitivity and specificity values. For example, a sensitivity of 100% for lung would indicate that all pixels within the image considered to be lung by the ground truth were detected as lung by the system. A specificity of 100% would indicate that all pixels identified as not belonging to lung by the ground truth were successfully excluded.

Table 3. Mean sensitivity (Sens.) and specificity (Spec.) as percentages for phases A, B and C

Test Set	A		B		C	
	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
Anatomy-20						
- Lungs	72.85	92.57	79.43	99.75	82.34	99.52
- Spine	40.90	99.81	88.28	99.59	88.28	99.59
- Sternum	7.58	99.95	40.72	99.85	46.98	99.79
- Shoulder	6.21	99.87	44.52	99.97	40.15	99.98
Lungs-40	88.40	99.80	93.90	99.70	96.10	99.40

The mean sensitivity and specificity for lungs, spine, sternum and shoulders across the three training phases are shown in Table 3. The high specificity values indicate that the processes are conservative in their detection and labeling of regions, hence less likely to include regions that are not lung, spine, sternum or shoulder. This is true for both Anatomy-20 and Lungs-40 datasets, with the mean specificity values around 99% across the three training phases. Only the lungs (in Anatomy-20) incorrectly include non-lungs regions during phase A, as indicated by a specificity of 92.57%, which was improved in subsequent training phases.

The sensitivity in segmenting lung regions saw an improvement across the three training phases for both the Anatomy-20 and the Lungs-40 datasets. The sensitivity for spine, sternum and shoulder regions improves significantly during phase B and remains relatively consistent during phase C. The sensitivity for shoulders (at 40.15%) and sternum (at 46.98%) remains quite low even during phase C, but this must be evaluated in context of their role within the system.

Firstly the intended purpose of shoulder and sternum regions is to acquire an approximation of the landmarks to correctly scale and align the statistical model used to segment lung regions. So even if these processes do not capture all of the pixels for

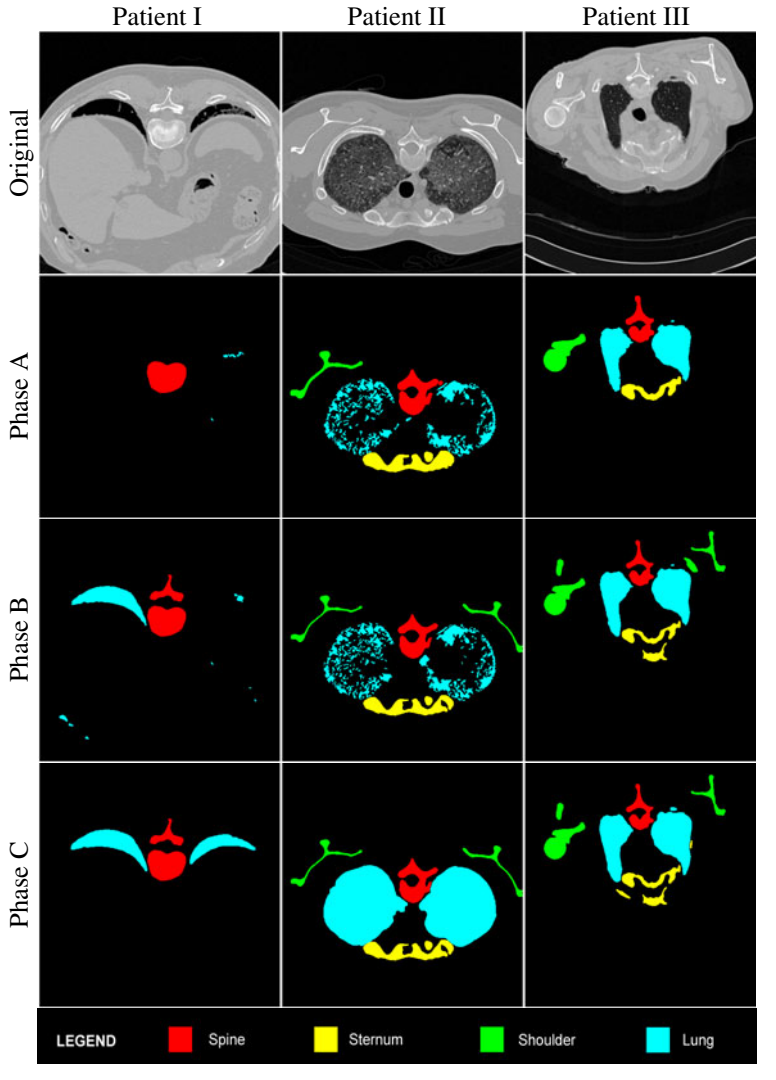


Fig. 4. Lung segmentation results. (Left to Right) Patient I, II and III. (Top to Bottom) Original HRCT scan, segmentation results after training phase A, B and C.

shoulder or sternum, even a partial segmentation is sufficient for their intended task. Secondly *PrevKnownShoulderArm* and *PrevKnownSternum* help to compensate for regions missed by *ShoulderArmDetection* and *SternumDetection* respectively, by retrieving the segmentation results for the preceding images for the same patient. Engineering a system as a network of processes allows limitations of an algorithm within one process to be compensated by other algorithms in other processes.

Three examples of segmentation results from the Anatomy-40 test set are shown in Figure 4. The top row represents the original HRCT image and subsequent rows

represent the segmentation results after training phase A, B and C, respectively. As mentioned earlier the training and test sets contain patients suffering from a variety of diffuse lung diseases. The lungs of patient I are affected by *emphysema* and *honeycombing* diseases patterns. The lungs of patient II are affected by *ground-glass-opacity*. Patient III has normal lung tissue, albeit suffers from *pleural plaques* affecting small parts of the lung wall.

At the end of training phase A, the anatomy segmentation system could correctly segment the healthy lung tissue as shown in patient III, but had difficulty segmenting lung regions in studies where the patient was severely affected by a variety of diffuse lung diseases in patient I and II. In training phase B, revisions to the system led to an improvement in its ability to segment lung regions in patients affected by emphysema, as observable by the improvements in results for patient I. It was also successful in segmenting shoulder regions missed previously by adjusting for the body's rotation as observed for patient II and III. By the end of training phase C, the system correctly segments lungs severely affected by honeycombing (patient I) and ground-glass-opacity (patient II). The segmentation for the sternum regions for patient III incorrectly includes other calcified regions and ribs. In patient II the sternum is fused with the ribs, thus making it difficult to separate them. Note that none of these cases from the test sets were used to train or revise the system. The improvements in these test cases were a result of improvements made on other training cases.

4 Discussion

It is reasonable to accept that when developing complex software systems, the required knowledge is often incomplete and further revisions are needed until the knowledge can converge to support the intended goals of the system. These incremental ad-hoc revisions carry the risk of degrading the system.

ProcessNet offers a strategy to facilitate incremental convergence in the knowledge of the system, irrespective of whether that knowledge is defined in rules or algorithms or other software systems. It allows experts the opportunity to identify poorly performing parts of the system and use evidence in the form of cornerstone cases to guide the revisions to improve them. Our anatomy segmentation system constructed using ProcessNet, demonstrates an improvement in the knowledge over three training phases as observed via improved segmentation results for lung, spine, sternum and shoulder regions on independent test sets.

The erroneous cases in the test set denote the system's current limitation, and if used to revise the system, will become cornerstones for the processes of the lung anatomy segmentation system. Once captured as cornerstones the validated change strategy of ProcessNet will ensure that any future changes to the system do not degrade the process or system's performance for these cases. Naturally the strength of such a system lies in the quality of its cornerstone cases. The relatively poor segmentation results for shoulder regions between phases B and C indicates that the current set of cornerstones did not cover all the variations present in the test set.

As the number of cases available grows over time, the expert can use pattern recognition and machine learning techniques to automatically derive the knowledge within the narrow scope of a process. The larger the pool of cornerstones cases, the

greater the confidence an expert can have in the quality of that process. Even though a ProcessNet system may have a large number of processes, the validated change strategy focuses the attention of an expert's revision on one process at a time. These qualities make ProcessNet scalable in the number of cases, but also in the number of processes within the system.

Further to this, multiple experts can engineer different parts of the system and only concentrate on improving the correctness of their specific processes. The validation of a process via its own cornerstones serves as a simple and effective approach to determine how the inputs to a process may change and adapt accordingly.

5 Conclusion

Incremental ad-hoc engineering of complex software systems, is an inevitable consequence of engineering systems as available data, expertise and techniques evolve. The risk of a change degrading the system's performance restricts our ability to build complex systems and confidently assimilate new data and techniques. In this work we have presented an incremental validated change strategy called ProcessNet that mitigates these risks. While ProcessNet may not eliminate all risk of degradations, the experiments support the idea that the accumulated set of cornerstones offer a data-driven means to assess the quality of a change to the system. This is the same argument and experience that underlies standard RDR.

ProcessNet captures and validates knowledge represented explicitly (such as in the form of rules) or implicitly in the libraries and algorithms defined in the source code of a process. The use of cornerstones at each process means that knowledge behind heuristically defined algorithms can be supported by evidence grounded in data. The medical image analysis system developed using ProcessNet, demonstrates that despite large numbers of ad-hoc revisions, an anatomy segmentation solution can be discovered and improved over time in a systematic manner.

References

1. Jain, A., Duin, R., Mao, J.: Statistical Pattern Recognition: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(1), 4–37 (2000)
2. Crevier, D., Lepage, R.: Knowledge-Based Image Understanding Systems: A Survey. Computer Vision and Image Understanding 67(2), 161–185 (1997)
3. Compton, P.J., Jansen, R.: A philosophical basis for knowledge acquisition. Knowledge Acquisition 2, 241–257 (1990)
4. Compton, P., Peters, L., Edwards, G., Lavers, T.: Experience with ripple-down rules. Knowledge Based Systems Journal 19(5), 356–362 (2006)
5. Bekmann, J., Hoffmann, A.: Improved Knowledge Acquisition for High-Performance Heuristic Search. In: International Joint Conference on Artificial Intelligence, pp. 41–46 (2005)
6. Misra, A., Sowmya, A., Compton, P.: Incremental Learning of Control Knowledge for Lung Boundary Extraction. In: Pacific Knowledge Acquisition Workshop, pp. 1–15 (2004)
7. Misra, A., Sowmya, A., Compton, P.: Incremental learning for segmentation in medical images. In: IEEE International Symposium on Biomedical Imaging (ISBI 2006). 1, pp. 1360–1363 (2006)

8. Compton, P., Cao, T., Kerr, J.: Generalising Incremental Knowledge Acquisition. In: Pacific Knowledge Acquisition Workshop, pp. 1–15 (2004)
9. Finlayson, A.: Incremental Knowledge-acquisition for Complex Multi-agent Environments. PhD Thesis, University of New South Wales, Sydney, Australia (2008)
10. Doi, K.: Current status and future potential of computer-aided diagnosis in medical imaging. The British Journal of Radiology 78, S3–S19 (2005)
11. Draper, B., Hanson, A., Riseman, E.: Knowledge-directed vision: control, learning, and integration. Proceedings of the IEEE 84(1-I) (November 1996)
12. Bovenkamp, E., Dijkstra, J., Bosch, J., Reiber, J.: Multi-agent segmentation of IVUS images. Pattern Recognition, 647–663 (2004)
13. Matsuyama, T.: Expert systems for image processing: Knowledge-based composition of image analysis processes. In: Proceedings of International Conference on Pattern Recognition, pp. 125–133 (1988)
14. Clouard, R., Elmoataz, A., Porquet, C., Revenu, M.: Borg: a knowledge-based system for automatic generation of image processing programs. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(2), 128–144 (1999)
15. Draper, B.A.: From knowledge bases to Markov models to PCA. In: Proceedings of Workshop on Computer Vision System Control Architectures, Graz, Austria (2003)
16. Draper, B.A., Bins, J., Baek, K.: ADORE: Adaptive Object Recognition. Videre 1(4), 86–99 (2000)
17. Levner, L., Bulitko, V.: Machine learning for adaptive image interpretation. In: Proceedings of The National Conference on Artificial Intelligence, pp. 890–876 (2004)
18. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: International Conference on Computer Vision (ICCV 2001), pp. 105–112 (2001)
19. El-Baz, A., GimelXfarb, G., Falk, R., Holland, T., Shaffer, T.: A New Stochastic Framework for Accurate Lung Segmentation. In: Proc. of International Conference on Medical Image Computing and Computer-Assisted Intervention, USA, pp. 322–330 (2008)
20. Massoptier, L., Misra, A., Sowmya, A.: Automatic Lung Segmentation in HRCT Images with Diffuse Parenchymal Lung Disease Using Graph-Cut. In: International Conference Image and Vision Computing New Zealand, pp. 266–270 (2009)
21. Brown, M.S., Wilson, L.S., Doust, B.D., Gill, R.W., Sun, C.: Knowledge-based method for segmentation and analysis of lung boundaries in chest X-ray images. Computerized Medical Imaging and Graphics 22(6), 463–477 (1998)
22. Sluimer, I., Schilham, A., Prokop, M., van Ginneken, B.: Computer analysis of computed tomography scans of the lung: a survey. IEEE Transactions on Medical Imaging 25(4), 385–405 (2006)
23. Rasband, W.S.: ImageJ, U. S. National Institutes of Health, Bethesda, Maryland, USA (1997-2009), `http://rsb.info.nih.gov/ij/`

RDRCE: Combining Machine Learning and Knowledge Acquisition

Han Xu and Achim Hoffmann

School of Computer Science and Engineering
University of New South Wales, Sydney, Australia

Abstract. We present a new interactive workbench RDRCE (RDR Case Explorer) to facilitate the combination of Machine Learning and manual Knowledge Acquisition for Natural Language Processing problems. We show how to use Brill's well regarded transformational learning approach and convert its results into an RDR tree. RDRCE then strongly guides the systematic inspection of the generated RDR tree in order to further refine and improve it by manually adding more rules. Furthermore, RDRCE also helps in quickly recognising potential noise in the training data and allows to deal with noise effectively. Finally, we present a first study using RDRCE to build a high-quality Part-of-Speech tagger for English. After some 60 hours of manual knowledge acquisition, we already exceed slightly the state-of-the art performance on unseen benchmark test data and the fruits of some 15 years of further research in learning methods for Part-of-Speech taggers.

Keywords: Knowledge Acquisition, Ripple Down Rules, Machine Learning, TBL, Part-of-Speech tagger.

1 Introduction

For long there has been a division of opinions regarding the best way of building a knowledge base. Manual approaches to build knowledge bases tend to be tedious while automatic approaches require usually large amounts of training data and the results may still be inferior to those achieved by manually building a knowledge base.

In this paper we present our new workbench Ripple Down Rules Case Explorer (RDRCE) which has been designed to facilitate both, the automatic construction of an initial knowledge base which is subsequently further improved in a manual fashion. This seems to be of particular value for applications in Natural Language Processing (NLP). Reason being that often some training data has been developed for machine learning, but the results of the applied learners are not really satisfactory as the NLP domains are too complex.

Ripple Down Rules [4] have proven to be a very effective way of manually building knowledge bases. The approach has been widely applied and adapted, including to various NLP applications, e.g. [8,11]. RDRCE not only allows to

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 165–179, 2010.

automatically generate an initial RDR tree using Transformation-Based Learning(TBL), which is a well-entrenched learning technique for certain NLP problems. It also interactively supports the inspection and refinement of the initial RDR tree to quickly improve upon the performance achieved by the machine learner. A common issue turned out to be that the training data may contain considerable degrees of noise. We designed RDRCE to support a user to quickly identify noisy training data and to correct it on the spot or mark it as dubious and to be set aside for later examination.

Finally, we applied RDRCE to the problem of Part-of-Speech tagging, which has attracted considerable interest by NLP researchers in the past. Since large scale training data has been available (i.e. more than 1,000,000 words with manually generated Part-of-Speech tags) virtually all approaches to build automatic Part-of-Speech taggers have been some sort of learning or statistical approach.

In our first case study we managed already to exceed slightly the best performance of any known Part-of-Speech tagger that has been used on the given training and test corpus with a total of about 60 hours of knowledge acquisition sessions. This is despite the fact that our training algorithm produces inferior results compared to the state-of-the-art. Furthermore, while developing that tagger, we also discovered that the widely used benchmark training and test data contains a considerable degree of noise.

The paper is organised as follows: The next section presents the background and related research. In section 3 we present the conceptual design of RDRCE. Section 4 contains a detailed description of RDRCE and its rule language. In section 5 we discuss our case study. Section 6 presents the conclusions and future work.

2 Background

2.1 Ripple Down Rules

Ripple Down Rules (RDR) [4] form the basis of our approach. RDR allows one to add rules to a knowledge base (KB) incrementally without the need of a knowledge engineer. A new rule is only created when the KB performs unsatisfactorily on a given case. The rule represents an explanation for why the conclusion should be different from the KB's conclusion on the case at hand. RDR was first used to build the expert system PEIRS for interpreting chemical pathology results [5].

A Single Classification Ripple Down Rules (SCRDR) tree is a binary tree where each node has between zero and two child nodes. The edges or links connecting the child nodes are typically called *except* and *if-not* (or *false*) edges. See Fig. 1. Associated with each node in a tree is a *rule*. A rule has the form: *if* COND *then* CONCLUSION where COND is called the *condition* and CONCLUSION the *conclusion*, which is most often simply a class label assigned to a case, but can be any kind of action. A *case* is an object to which the RDR knowledge base is applied, such as an attribute-value vector that should be classified. To decide on the conclusion to be applied to a case given an SCRDR tree we start

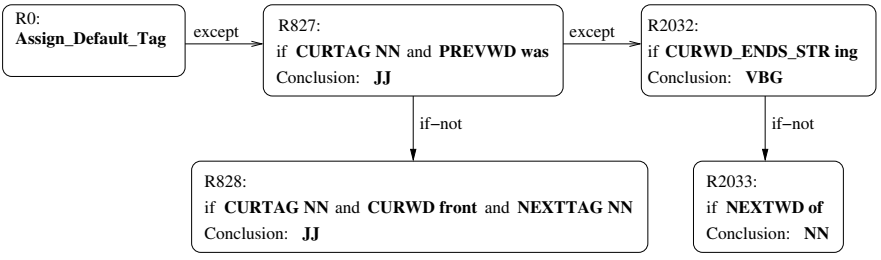


Fig. 1. Part of a Ripple Down Rules tree for Part-of-Speech tagging

with the root node. At each node that is encountered, the corresponding rule condition is evaluated for the case. If the case satisfies the condition, we say that the node *fires*. If a node fires, we follow the exception link of that node, if there is any. Otherwise, we follow the if-not link, if there is any.

The final conclusion performed 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 performed, the root node typically contains a trivial condition which is always satisfied. This node is called the *default* node. A new node is added to an SCRDR tree when the evaluation process returns the wrong conclusion. The new node is attached to the last node in the evaluation path of the given case. If that node fired, the new node is attached via an *except* link and otherwise via an *if-not* link.

Besides SCRDR a number of other styles of Ripple Down Rules have been developed which are useful for specific application domains, such as Multiple Classification RDR (MCRDR) [7], or [10] to just name a few. An excellent recent survey paper on the RDR methodology is [14]. In the following we will limit our consideration to SCRDR trees, even so we would just speak of RDR trees for brevity.

2.2 Part-of-Speech Tagging

Part-of-Speech (POS) Tagging refers to the process that assigns each word in a natural language sentence with a suitable tag from a given set of tags, such as Verb, Noun, etc. It has its significance in various areas of Natural Language Processing (NLP), such as Parsing, Machine Translation, Text to Speech and Information Extraction. It is usually used as a low-level pre-processor for other high-level NLP tasks, thus its accuracy has a direct influence on the final performance of the entire system. Various automatic POS tagging techniques and applications have been developed over the past few decades. Early approaches were hand-crafted rules, e.g. [9]. The Brill tagger, e.g. [1], using a machine learning approach resulted in a higher accuracy than those earlier attempts. Since then a considerable number of more sophisticated statistical and machine learning approaches have been developed to improve on the performance of the Brill tagger. Those developments elaborate on various statistical techniques, such as

Hidden Markov Models, e.g. [3], or Maximum Entropy, e.g. [13]. The most recent advancements in statistical POS tagging using a very elaborate technique involving a Perceptron [18] have pushed the state-of-the-art tagging accuracy on the Penn Treebank corpus Wall Street Journal to 97.44%.

3 Our Approach

Our aim is to build high quality knowledge bases for domains where there is a considerable amount of classified training data available. We developed our RDR case Explorer (RDRCE) which allows to automatically generate an initial RDR tree which can subsequently be further refined and improved by supporting the browsing of existing training data as well as it allows to look at new data as it becomes available. However, since training data is often prone to noise, RDRCE has also been designed to allow the efficient checking of suspicious training data and allows to record data that has been identified as noisy or that warrants more thorough checking at a later stage.

If a large volume of training data is available it appears sensible to take advantage of that and build automatically an initial RDR tree and to improve and refine it subsequently by interaction with a domain expert. Building a sophisticated knowledge base with our approach can be divided into three phases:

1. Generating the initial RDR tree. If classified training data is available, RDRCE uses Transformation-Based Learning (TBL) [1] to generate a rule list which can be automatically converted into an RDR tree.

   If no training data is available or if one opts to build the RDR tree from scratch, the RDR tree would only consist of the root node and a default conclusion.
2. Continued refinement of the initial RDR tree: RDRCE allows the user to gain quickly an in-depth understanding of the domain and the short-comings of the rules generated so far. Furthermore, as some of the training data may also be incorrectly classified, RDRCE also allows the user to correct the classification of training cases or to mark them as dubious for detailed checks at a later stage.

   A *case* in the training or testing corpus is a token (usually a word) along with its provided tag. Each case is provided with its context, i.e. words and their respective tags on both sides of the tag in question. We call the *cornerstone cases of a rule* r all cases in the training corpus for which r fires and assigns the correct tag, i.e. the same tag as provided in the training corpus. We call *bad cases* of a rule r all cases in the training corpus for which r fires but assigns an incorrect tag, i.e. different to the tag in the training corpus.
3. Adding new RDR rules: RDRCE provides a more versatile rule language than the rule templates used in Brill's tagger. It also allows to build dictionaries which hold an extensible list of words or tags. Such dictionaries can be used in rule conditions where a given word or tag in a case is tested for being a member in a given dictionary.

Quite a few learning algorithms have been developed which either directly build RDR trees from classified data, e.g. INDUCT [6], or which build similar structures, such as decision trees that can be transformed into an RDR tree. Investigations into the combination of ML and RDR had also been carried out by researchers in the past. Wada et al. [21] demonstrated a flexible knowledge-based system, where the KB is constructed by a human expert using RDR at an earlier stage of the development when there is no large amount of data available. Subsequently, when sufficient data is available, the KB is refined by using a Minimum Description Length Principle based induction technique. Suryanto and Compton [19] presented a method to minimise the KA work load for the human expert by enabling the RDR system to generalise the knowledge acquired from the expert and to automatically discover intermediate conclusions. Other studies of using RDR exception structures for Machine Learning include Catlett [2] and Scheffer [16].

For our study, we chose to build on Brill's TBL approach [1] which proved to perform exceptionally well in the domain of Part-of-Speech tagging. Brill's TBL approach has also been successfully applied to solve many other problems, including Noun Phrase Chunking [12] and Dialog Act Tagging [15]. Furthermore, it appears to provide an excellent basis for an initial RDR tree which can then be further improved in a manual fashion.

In the following we describe Brill's transformational learning approach and show how the learning result can be converted to an RDR tree.

3.1 Turning the Brill Tagger into an Initial RDR Tree

As mentioned in section 2.2, Brill developed his Transofrmation-Based Learning (TBL) [1] approach for automatically acquiring rules for a rule-based tagger from a manually tagged training corpus.

Initially, known words are tagged with their most-frequent tag and unknown words are tagged either as 'noun' or 'proper noun', depending on whether the word is capitalized. Tagging errors are identified by comparing the current tagging with the training corpus and candidate error-correction rules are considered. The candidate rules are constructed by exhaustively instantiating a set of rule templates, which make reference to a token's lexical features (morphological features such as a token's prefix or suffix) and contextual features (such as the token's preceding/following token or POS tag). All rules are of the form FROM_TAG TO_TAG if COND. For lexical rules, COND may include predicates like 'the word has suffix XY'. For contextual rules, COND may include predicates like the surrounding tags are X and Y. Only the rule candidate with the highest net improvement is added to the rule list. This newly learned rule is subsequently applied to the whole training corpus and effects changes where the rule's condition is satisfied. The same process to add the next rule is repeated until there is no rule whose net improvement reaches a user-specified threshold.

The outcome of the rule learning process is an ordered list of transformation rules. We converted the so generated rule list automatically into an RDR tree as follows:

3.2 Converting a TBL Rule List Into an RDR Tree

Following is the algorithm RLRDR (Rule List to Ripple Down Rules) which converts a TBL rule list to an RDR tree:

```
rdrnode RLRDR(corpus C, rule_list R) {
 LOOP:
    if (length(R) == 0) return NULL;
    else {
        Rule = pop(R); // remove the lowest indexed rule from R
        R_e = R - Rule;
        R_f = R - Rule; //R_e = R_f = rest of the rule list
        for (case in C) {
            if (Rule is evaluated to True) {
                apply_rule(Rule, case);
                C_e.append(case);
            } else C_f.append(case);
        }
        if (length(C_e) != 0) { //if the Rule applied at least once to corpus C
            N = new rdrnode; N→question = Rule;
            N→cornerstonecases=set_correctly_classified_cases;
            N→except = RLRDR(C_e, R_e); N→false = RLRDR(C_f, R_f);
            return N;
        } else goto LOOP;
    }
}
```

4 Ripple Down Rules Case Explorer (RDRCE)

4.1 RDRCE's Rule Language

Our rule language builds upon the rule templates used in the Brill Tagger [1], but goes significantly beyond that. For a full list of the rule templates used in the Brill tagger, see [1].

We follow the notation of the rule templates by Brill. As a consequence, each rule consists of two parts – the action part and the condition part. The action part itself consists of two tags, the first one being the current tag of a token and the second tag being the new tag assigned to the token, if the rule condition is satisfied. I.e. each rule is of the following form:

FROM_TAG TO_TAG CONDITION

The rule condition can be a single elementary condition or a combination of such using *and*, *or*, and *negation*. In our rule language, we added the following new elementary condition types to those already available in Brill's rule templates. Our extensions can be divided into the following groups:

1. Dictionary membership checks: checks if a token or tag at a specific relative position to the focus token is in a predefined group (dictionary). The following condition names are available with their intuitive meaning. Each condition name needs to be followed by the name of the respective dictionary. E.g. CURWD_IN_DIC BE_DIC would require the current token to be some form of *be*. The other conditions available are the following: PREV1OR2OR3WD_IN_DIC, PREV1OR2WD_IN_DIC, PREV2WD_IN_DIC, PREVWD_IN_DIC, CURWD_IN_DIC, NEXTWD_IN_DIC, NEXT2WD_IN_DIC, NEXT1OR2WD_IN_DIC, and NEXT1OR2OR3WD_IN_DIC. Furthermore, all conditions above can also check whether the assigned tag to the respective token is in the specified dictionary. The condition names are PREVTAG_IN_DIC etc. During the knowledge acquisition process using RDRCE, we developed dictionaries that contained the following word or tag groups: all forms of *be*, *have*, or *do* respectively. All variants of adjective, noun, adverb, and verb tags. Others include time markers and causative words, such as *help*, *let*, etc.

   The purpose of introducing those dictionaries is to raise the levels of abstraction in a rule's condition to make the rules expressive enough to generalize well to unseen cases while keeping the rules concise and easily interpretable. All of those dictionaries are constructed incrementally along the way when doing the knowledge acquisition. Our workbench allows the user to directly interact with those dictionaries at run time. If the user encounters a new word it can immediately be added to a suitable dictionary.
2. Additional elementary conditions:
   - NEARBYTAG (NEARBYWD) – checks if a user specified tag (word) is on either side of the focus token
   - SURROUNDWD – checks if the focus token is in the middle of the two user specified words
   - PREVBIGRAMWD (NEXTBIGRAMWD) – checks if the preceding (following) bigram is the user specified bigram
3. Elementary lexical conditions: they check the following lexical properties of a token:
   - ISTITLE – checks if the focus word starts with a capital letter. ISLOWER (ISUPPER) – checks if the focus word is without any (with only) upper case letter. CURWD_IS_DIGIT – checks if it is a digit
   - HAS_STR *str* – checks if the focus word contains *str* as a substring. CURWD_ENDS_STR *str* (PREVWD_ENDS_STR *str*) – checks if the focus word (previous word) ends with *str*.

4.2 Inspecting and Refining RDR Trees with RDRCE

RDRCE provides various useful statistics, such as reports on the type and size of error clusters. By *error cluster* T_1-T_2, we mean all cases that are tagged T_1 by the knowledge base but the training data tag is T_2. RDRCE also provides the current tagging accuracy for a given tag as well as for a prospective rule which the user might consider to commit to the knowledge base. Error distributions are automatically calculated and provided to the user. The user can freely navigate

through the existing RDR tree by choosing a specific node and traversing the tree or by choosing a specific error cluster.

To understand why certain errors are being made by the knowledge base the user can view the cornerstone cases and the bad cases associated with a given rule. Since the number of cases can be quite large (several hundred cases is quite normal) various sorting and filtering functions (such as filtering by nearby contexts) can be used for displaying purposes. This allows an easier comparison between cornerstone cases and bad cases in order to develop valid refinement rules. Once a new rule is formulated the user can test its performance immediately on the existing corpus to assist in deciding to commit the rule to the knowledge base. (This is different from the traditional RDR approach, where cases are presented to the user one at a time. A rule is added to fix this single case, without having a good impression on how well the rule might generalize to other forthcoming cases.) Fig. 2 shows a screenshot of RDRCE displaying the performance of a candidate rule.

Fig. 2. This screenshot shows how the performance of the candidate rule EX RB NEXT1OR2OR3WD_IN_DIC BE_VB_DIC on the training corpus can be evaluated before the rule is actually committed

5 Case Study

5.1 Building the Initial RDR Tree

We used the Penn Treebank 3 (PTB) corpus and we use the same data division of the PTB parsed section as all of [3,20,17,18] do in order to allow better comparisons. We trained the Brill tagger V1.14 to obtain our initial rule lists. This resulted in a learned lexical rule list of length 200 and a contextual rule

list of size 932. We converted the 932 contextual rules into our initial RDR tree using our automatic conversion algorithm.

This resulted in 1,318 nodes for the initial RDR tree (some TBL rules resulted in multiple nodes in the RDR tree). The tagging process then used first the lexical rules and subsequently the contextual RDR rule tree. This resulted in an accuracy of 97.31% on the training corpus. An accuracy of 97.08% was achieved on the previously unseen sections 22-24 of test data. In this study we limited ourselves to acquiring contextual rules, as they appeared to hold more potential for improvement.

5.2 Inspecting the Initial RDR Tree

There are some nodes in the initial RDR tree that have a negative effect. This is as a consequence of the TBL calculating scores for candidate rules on the entire training corpus. In contrast, a TBL rule may have multiple copies in the initial RDR tree each applicable only to a local context, i.e. a small subset of the entire training corpus. So, the performance of the various copies of a rule can vary considerably. RDRCE allows to specifically analyse and fix those rules that have a negative score in the local RDR tree context. This might quite possibly result in cancelling the rule in its entirety. Here is such an example:

In node #1011, the rule: RB JJ LBIGRAM how much yields a negative score on bad cases like the following:

1) But/CC how/WRB **much/JJ** will/MD shoppers/NNS benefit/VB
2) The/DT question/NN remains/VBZ :/: how/WRB **much/JJ** can/MD the/DT West/JJ

There is no cornerstone case for this particular node, i.e. no case where the rule produces a correct result. The rule was generated in the TBL learning process because in other contexts there were cases for which the rule produces a correct result. RDRCE allows to retrieve those supporting cases from other nodes in the initial RDR tree to provide a better understanding in what circumstances the rule makes sense. For the above rule the cornerstone cases for the nodes #378, #660, #1201 and #1217 are collected and displayed by RDRCE. Below are some of those cornerstone cases given as examples:

decided/VBN when/WRB or/CC by/IN how/WRB **much/JJ** ./.
did/VBD n't/RB say/VB by/IN how/WRB **much/JJ** ./.
,/, showing/VBG traffic/NN wardens/NNS how/WRB **much/JJ** time/NN the/DT motorist/NN

In all of the bad cases, there is a modal (MD) following the phrase "how much". It is most likely, that a verb is somewhere in the right contexts which the "much" is modifying, thus the "much" should be tagged as a RB rather than JJ. This is clearly visible for the first bad case 1) above: the "much" is modifying the verb "benefit", but the window is not wide enough for the second bad case. The user can request a full sentence inspection which shows the following for 2):

> "/" The/DT question/NN remains/VBZ :/: how/WRB **much/JJ** can/MD the/DT West/JJ German/JJ market/NN absorb/VB ?/. "/" says/VBZ one/CD senior/JJ dealer/NN ./.

Now it becomes clear that the "much" in this case is modifying the verb "absorb" 6 positions to the right of the focus token "much". Thus, a following MD is a good cue that "much" in the preceding phrase "how much" should be tagged as adverb (RB). This leads to the addition of the following exception rule: JJ RB NEXTTAG MD. I.e. change the tag from adjective (JJ) to adverb (RB), if the immediately following tag is a modal (MD).

Other rules were only learned by the TBL due to incorrect manual tagging of the training corpus. Below are some examples:

Node #286 with the rule: VBZ NNS PREVTAG IN had a negative score of -22 given its local context in the generated RDR tree. The rule says: change the tag from verb 3rd person singular present (VBZ) to noun plural (NNS) if the previous tag is preposition (IN). All the bad cases of this node are about the token *'s* as shown in the following examples:

> allowed/VBN in/IN there/RB ,/, that/IN **'s/NNS** all/DT I/PRP know/VBP But/CC he/PRP says/VBZ that/IN **'s/NNS** no/DT more/JJR a/DT requirements/NNS ,/, and/CC that/IN **'s/NNS** a/DT source/NN of/IN

Clearly, *'s* should never have the possible tag of NNS (noun plural). Inspecting the training corpus shows that the token *'s* had been erroneously tagged as NNS while it should be VBZ, as e.g. in:

> "/" I/PRP 've/VBP never/RB seen/VBN so/IN many/JJ New/NNP York/NNP City/NNP G.O./NNP **'s/NNS** up/IN for/IN sale/NN ,/, "/" said/VBD another/DT trader/NN ./.

So we added the following wild-card rule to effectively cancel the rule and to revert all of the bad cases within this node: NNS VBZ ANY ANY.

Node #286 is an exception rule to node #284. The rule in #284 is: POS VBZ NEXTTAG DT, which allows only cases with ending *'s*, due to the fact that *'s* is the only token that can be tagged as both POS (possessive ending) and VBZ. Based on that the expert can have a high degree of confidence that the cancellation of the rule in node #286 is correct.

Another interesting situation was found in node #1161, which covers 14 cases while yielding 0 improvement. The node's rule is VBD VBN WDNEXTTAG received IN. I.e. change tag from verb past tense (VBD) to verb past participle (VBN), if the focus word is *received* followed by a preposition (IN). E.g.

> The/DT funds/NNS **received/VBN** from/IN pharmaceutical/JJ firms/NNS the/DT value/NN of/IN the/DT stock/NN **received/VBN** in/IN a/DT conversion/NN

Bad cases within this node include the following examples:

> dating/VBG back/RB to/TO forms/NNS it/PRP **received/VBN** in/IN 1985/CD ./.
> dividend/NN similar/JJ to/TO one/CD he/PRP **received/VBN** before/IN selling/VBG his/PRP$

It is fairly easy to notice the difference here: for the cornerstone cases, the focus word is following an NN or NNS, which results in the phrase structure: "*something* received IN", where it is very likely that "received" here is indeed used in a passive voice. However, for the bad cases, the tag preceding "received" is a personal pronoun which resulted the following structure: "*someone* received IN", where it is more likely that the focus word is immediately following its subject, and thus, is used in active voice and should be tagged as VBD rather than VBN. Based on this observation, the following rule was added as an exception rule to this node: VBN VBD PREVTAG PRP. Fig. 3 shows a screenshot when the above rule is tested.

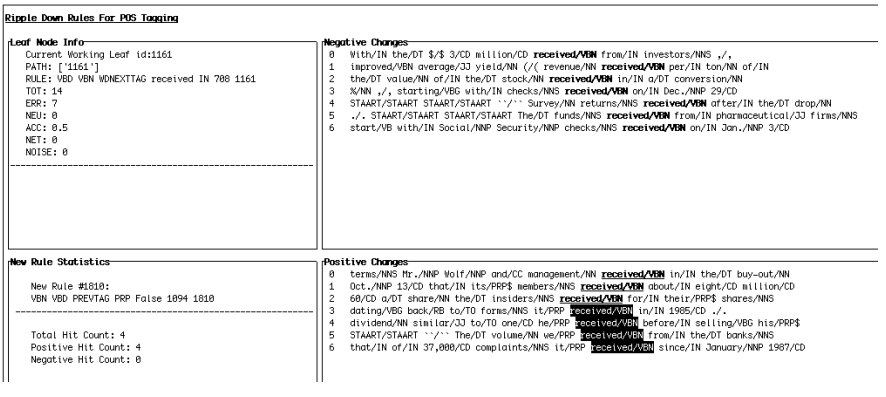


Fig. 3. Screenshot of RDRCE examining the performance of the new rule VBN VBD PREVTAG PRP

5.3 Adding Rules for the Residue

The set of training cases which are not retagged by the TBL is often called the *residue*. Each of the 912,344 training tokens were initially tagged by the start-state-tagger (part of Brill tagger, which simply assigns the most frequently occurred tag to a given token). Only 39,139 (4.29%) tokens were retagged by the rules produced by the TBL, or equivalently, by our initial RDR tree. The residue of 95.71% does not need any retagging in most cases, i.e. 94.48% are actually correctly tagged. This leaves 1.33% (or 11,237 cases) of the training corpus in need of retagging, which actually makes up some 46% of all erroneous tags. All of the *residue* is in the last node along the *if-not* link chain of the initial RDR tree.

In the residue, RDRCE found 251 error clusters. 144 out of those have a size ≤ 10. For our study, we were mainly interested in the larger error clusters with size ≥ 50, where a potentially higher performance improvement can be achieved. Adding rules to cope with tagging errors in the residue is as described in the previous section, except that the navigation is based on error cluster names rather than node numbers. Below is an example:

In the error cluster: "EX-RB" (tagged as "existential there" (EX) but should be adverb (RB)), the following rule was added: EX RB NEXT1OR2OR3WD_IN_DIC BE_VB_DIC based on a very simple intuition that the *existential there* should usually be followed closely in its right context by some form of *be*. If there is no form of *be* following, the correct tag of "there" will more likely be adverb (RB). This single rule fixed 34 (54%) cases within this error cluster. See also Fig. 2.

However, not all errors in the residue are this easily addressable. Our experience showed that the larger the error cluster the harder it is to formulate rules. This is because it is more difficult to tailor rules to accommodate a large set of cases, even if one allows for a reasonable number of exception rules. However, developing rules for the residue clusters was not only harder because of the size of the clusters but also because of the fact that the TBL did not manage to come up with applicable rules, thus indicating that the individual cases themselves are quite difficult to discern.

5.4 Discovering Noise in the Training Data

RDRCE made it relatively easy to detect noise in the training data and also allowed the user to correct the data by the press of a single button for each case. Much of the noise was also found in the residue. It seems likely that due the noise the cases ended up in the residue, as the TBL was incapable of finding effective rules.

For example, the PTB tagging guidelines on page 12 state: "Hyphenated modifiers, should always be tagged as adjectives (JJ)". However, in the error cluster NN-JJ we find e.g. that "junk-bond" in the phrase "junk-bond market" was tagged as NN 15 times and as JJ 4 times although the tagging guidelines say clearly that it should be tagged as JJ in all cases. Further, when we tried out the following rule within error cluster NN-JJ: NN JJ HASSTR '-' and NEXTTAG NN or NNS, which is expressing the above guideline in our rule language, the new rule gave us 229 positive changes but 1255 negative changes. As a next step using RDRCE we can walk through the 1255 negative cases and can verify whether all of them are noise. While the number of cases is large, it appears to take only a fraction of a second for most cases to verify that they are noise indeed.

5.5 Results

While we discovered a considerable amount of noise in the training data we did not correct it in order to allow a sensible comparison of the performance of our resulting RDR tree with the best of other POS taggers. Where available, we report the accuracies on both, the training data (WSJ section 0-18) as well as

the unseen test data (WSJ section 22-24). While the sections 19-21 were used by others for calibrating their learning approaches, we did not need that and did not touch those sections for comparison purposes. The respective results are shown in Table 1. We spent about 60 hours of knowledge acquisition to refine the initial RDR tree. During the knowledge acquisition process we added 415 rules to the initial RDR tree. This resulted in a performance exceeding the best known POS tagger to date [18], which as automatically learned, though. However, it took some 15 extra years of research to develop the algorithm used in [18] compared to Brill's TBL learner which we used as a basis for our POS tagger development. We think that the considerable effort that went into developing the today's best learning techniques, e.g. in [18], would well exceed the manual effort required to refine an initial RDR tree using RDRCE. Of course, we spent also quite an effort for building RDRCE. However, we believe that RDRCE can be easily applied to other tasks in the Natural language Processing domain, such as citation classification, sentence classification, or information extraction whenever there is a training corpus available.

Table 1. Part-of-Speech tagging results for the PTB WSJ corpus. The knowledge base built using RDRCE within some 60 hours exceeds, albeit slightly, the performance of the best automatically learned taggers to date.

Tagger	Accuracy Training WSJ00-18	Accuracy Test WSJ22-24
Brill's Tagger (turned into inital RDR tree)	97.31%	97.08%
Ratnaparkhi 1996	unknown	96.63%
Collins 2002	unknown	97.11%
Toutanova et al. 2003	unknown	97.24%
Shen et al. 2007	unknown	97.33%
Spoustova et al. 2009	unknown	97.44%
RDRCE	97.76%	**97.46%**

6 Conclusions and Future Work

We presented our new RDRCE which was designed to support the hybrid process of building an RDR knowledge base partly manually and partly automatically using available training data.

Using RDRCE we showed that it is feasible to develop a Part-of-Speech tagger that exceeds, albeit slightly, the current state-of-the-art on standard test data within a week and a half or so of knowledge acquisition sessions. The manual effort spent appears to compare quite favourably to the effort that was spent otherwise on improving the state-of-the-art of learning algorithms for part-of-speech tagging. Overall, it appears that the best possible performance for any Part-of-Speech tagger will lie well below 100%. This is partly due to noise in the underlying corpus, as we discovered, but also partly due to the fact that even humans will disagree on some tags.

We employed Brill's TBL approach to generate a transformation rule list. We introduced a new conversion algorithm in section 3.2 that turns the transformation rule list into an RDR tree and assigns the training data to the relevant nodes in the RDR tree. We believe that the TBL approach is applicable not only to the problem of Part-of-Speech tagging but also to a host other NLP problems, including citation or sentence classification, or information extraction. Our initial comparison of the performance of the TBL with alternative learning algorithms whose results is either an RDR tree or can be converted into one, namely INDUCT and C4.5, favoured strongly the TBL approach. For the full training data set, the WEKA implementations of the named algorithms did not actually manage to complete the learning as they ran out of memory. It is also not clear how to take the flexibility of the TBL into account, as it allows to adapt it easily to a new problem domain by modifying the templates it uses. This kind of domain knowledge appears to be more difficult to provide to the other learners in a feasible way (as they tend to run into computational problems with too many features).

However, future work should look more closely into the comparative performance of different suitable learning algorithms. Furthermore, we intend to explore to what degree the refinement of an initial RDR tree can be further automated.

References

1. Brill, E.: Some advances in transformation-based part of speech tagging. In: AAAI 1994: Proceedings of the Twelfth National Conference on Artificial Intelligence, vol. 1, pp. 722–727 (1994)
2. Catlett, J.: Ripple-down rules as a mediating representation in interactive induction. In: Proceedings of the Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop, Kobe, Japan, pp. 155–170 (1992)
3. Collins, M.: Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In: EMNLP 2002: Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing, p. 10 (July 2002)
4. Compton, P., Jansen, R.: A philosophical basis for knowledge acquisition. Knowl. Acquis. 2(3), 241–257 (1990)
5. Edwards, G., Compton, P., Malor, R., Srinivasan, A., Lazarus, L.: Peirs: a pathologist maintained expert system for the interpretation of chemical pathology reports. Pathology 25, 27–34 (1993)
6. Gaines, B.R.: An ounce of knowledge is worth a ton of data: Quantiative studies of the trade-off between expertise and data based on statistically well-founded empirical induction. In: Proceedings of the 6th International Workshop on Machine Learning, pp. 156–159 (June 1989)
7. Kang, B., Compton, P., Preston, P.: Multiple classification ripple down rules: Evaluation and possibilities. In: Proceedings of the 9th AAAI-sponsored Banff Knowledge Acquisition for Knowledge Based Systems Workshop, pp. 17.1–17.20 (1995)
8. Kim, Y.S., Kang, B.H., Choi, Y.J.: Incremental Knowledge Management of Web Community Groups on Web Portals. In: 5th International Conference on Practical Aspects of Knowledge Management, Vienna, Austria, pp. 198–207 (2004)

9. Klein, S., Simmons, R.F.: A computational approach to grammatical coding of english words. ACM 10(3), 334–347 (1963)
10. Martinez-Bejar, R., Ibanez-Cruz, F., Le-Gia, T., Cao, T.M., Compton, P.: Fmr: An incremental knowledge acquisition system for fuzzy domains. In: Fensel, D., Studer, R. (eds.) EKAW 1999. LNCS (LNAI), vol. 1621, pp. 349–354. Springer, Heidelberg (1999)
11. Pham, S.B., Hoffmann, A.: Efficient knowledge acquisition for extracting temporal relations. In: Proceedings of the European Conference on Artificial Intelligence (ECAI), Riva del Garda, Italy, pp. 521–525 (2006)
12. Ramshaw, L.A., Marcus, M.P.: Text chunking using transformation-based learning. In: Proceedings of the Third Workshop on Very Large Corpora, pp. 82–94 (1995)
13. Ratnaparkhi, A.: A maximum entropy model for part-of-speech tagging. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, vol. 1, pp. 133–142 (1996)
14. Richards, D.: Two decades of ripple down rules research. The Knowledge Engineering Review 24(2), 159–184 (2009)
15. Samuel, K., Carberry, S., Vijay-Shanker, K.: Dialogue act tagging with transformation-based learning. In: Proceedings of the 17th International Conference on Computational Linguistics (August 1998)
16. Scheffer, T.: Algebraic foundations and improved methods of induction or ripple-down rules. In: Proceedings of the 2nd Pacific Rim Knowledge Acquisition Workshop, Sydney, Australia, pp. 279–292 (1996), ISBN: 0-7334-1450-8
17. Shen, L., Satta, G., Joshi, A.K.: Guided learning for bidirectional sequence classification. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 760–767 (June 2007)
18. Spoustová, D., Hajič, J., Raab, J., Spousta, M.: Semi-supervised training for the averaged perceptron pos tagger. In: EACL 2009: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (March 2009)
19. Suryanto, H., Compton, P.: Invented predicates to reduce knowledge acquisition. In: Motta, E., Shadbolt, N.R., Stutt, A., Gibbins, N. (eds.) EKAW 2004. LNCS (LNAI), vol. 3257, pp. 293–306. Springer, Heidelberg (2004)
20. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, pp. 173–180 (2003)
21. Wada, T., Motoda, H., Washio, T.: Knowledge acquisition from both human expert and data. In: Cheung, D., Williams, G.J., Li, Q. (eds.) PAKDD 2001. LNCS (LNAI), vol. 2035, pp. 550–561. Springer, Heidelberg (2001)

Simulated Assessment of Ripple Round Rules

Ivan Bindoff and Byeong Ho Kang

University of Tasmania, School of Computing and Information Systems
{Ivan.Bindoff,bhkang}@utas.edu.au

Abstract. A new Ripple Down Rules based methodology which allows for the creation of rules that use classifications as conditions has been developed, and is entitled Multiple Classification Ripple Round Rules (MCRRR). Since it is difficult to recruit human experts in domains which are appropriate for testing this kind of method, simulated evaluation has been employed. This paper presents a simulated evaluation approach for assessing two separate aspects of the MCRRR method, which have been identified as potential areas of weakness. Namely, "Is the method useful in practice?" and "Is the method acceptable, computationally?" It was found that the method appears to be of value in some, but not many, "traditional" multi-class domains, and that due to computational concerns with one aspect of the method it is considered unsuitable for domains with a very large number of cases or rules. These issues are discussed and solutions are proposed.

Keywords: ripple, down, rules, multiple, classification, round, configuration, knowledge acquisition, simulated, expert, assessment.

1 Introduction and Previous Work

Knowledge based systems are notoriously difficult to evaluate objectively for several reasons. Particularly, it is difficult to get experts to give their time to the training of the project and it is virtually impossible to make them train the same system multiple times. Furthermore, it is difficult to get a true gauge of how correct or optimal the system really is, as it is difficult to convince the expert or other independent experts to verify the results of the system after the fact. To compound these problems further still, different experts will have different opinions as to what correct is, and are likely to perform somewhat inconsistently from one train of a system to the next. Because of these features, multiple experts should really be assessed and contrasted multiple times each. Since experts are - virtually by definition - scarce, and their time valuable, this kind of assessment is very rare [1]. To overcome this problem, simulated experts have been previously employed for the task of evaluating new expert systems methodologies, particularly in Ripple Down Rules (RDR) related studies, such as this one [2-4].

Having developed a new RDR based method, and having very limited access to human experts with appropriate domains to evaluate, it was considered necessary to again employ simulated experts to evaluate the method. However, the particular

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 180–194, 2010.

characteristics of the method required some modifications to the existing simulated expert evaluation process.

2 Ripple Down Rules

The RDR methodology makes use of a true-false binary tree structure in order to ensure that rules are always added in context [5, 6]. An example of this structure is shown in Figure 1.

Using this same sample knowledge base we can also consider the inference process that is used in the RDR methodology. If we consider a sample case in which we have [X=5, Y=5, Z=10] then we will consider rule 1 initially and find that X>4 is true, so rule 2 will be considered. Y<3 is false, so rule 7 must be considered. The condition Z<9 is also false, but as there is no false branch deriving from rule 7 we have hit a dead end and as such will simply fire the last considered true statement, which results in rule 1 firing [4].

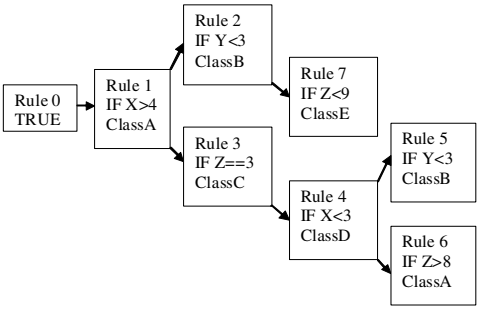


Fig. 1. A sample RDR knowledge base where arrows pointing upwards indicate the TRUE path while arrows heading downwards indicate FALSE paths

RDR, as described thus far, still encounters problems with maintainability. The addition of a new rule may cause previously considered cases to become misclassified, however, this problem was easily solved. When creating a new rule, the system would store the current case against that new rule as a "cornerstone" case, a copy of the case as it existed when the rule was created. Later, when creating further rules, the system will detect if this new rule caused a conflict with the past cornerstone. That is, did the new rule change the cornerstone's classification? If so, then at this stage the expert was required to select a relevant difference between the current case and the cornerstone case [7, 8]; this process is often called the validation and verification stage of RDR.

The RDR method as described is unsuitable for multiple classification tasks, since it would require the use of either multiple knowledge bases or compound classifications, which can cause an undesirable explosion in the amount of knowledge required [1].

To extend this method to multiple classification tasks, Kang altered the underlying knowledge representation structure to that of an n-tree, altered the cornerstone case

approach such that multiple cornerstone cases might apply to a single case, and modified the inference strategy such that it would not cease operation when a correct classification was found, but instead would add any deepest-satisfied node to the result set [4]. This method was entitled Multiple Classification Ripple Down Rules (MCRDR).

RDR methods, unlike many proposed expert system methodologies, have proven to be quite valuable in real world tasks, with one very successful commercial system for pathology interpretation [9], another for tesco.com [10], and several commercial applications under development in areas including high volume call centre management [11] and medication review [12].

2.1 Using Classifications as Conditions

One of the compromises that were made when developing the RDR methodology, when compared to some traditional expert systems methodologies, was that the ability to create rules which used classifications as conditions was lost. This ability had previously been integral to success in some domains, and is of particular value in complex configuration and planning tasks where the positioning of each module/event may influence the positioning of other modules/events.

This shortcoming was recognized quite early, with Mulholland developing an RDR based system to solve Ion Chromatography configuration tasks, although this solution was highly specialized and would require extensive redesign in order to be applied in other domains [13]. It was also unreliable, since it was possible to create rules which would cause infinite cycles, which necessitated a program halt, and had to be removed manually after they were observed [13]. After this, Beydoun & Hoffman developed Nested RDR, a single classification RDR approach which allowed the creation of "intermediate classifications", as stepping stones towards the end classification [14]. This method was more generally applicable than Mulholland's offering, being not targeted at a single, specific domain, but was targeted primarily at single classification problem domains, with intermediate classifications being treated separately to true classifications [14]. Later, a proposition was made for a more generalized version of RDR, which included provision for a Repeat Inference MCRDR (RIMCRDR) approach whereby the existing MCRDR method was augmented with the ability to use classifications as conditions [15, 16]. However, this approach included many restrictions as to when and how these types of rules could be used, and about how the knowledge base must be inferred and interpreted. This was done in an attempt to eliminate the potential for cyclic rules – rules which depend on the existence of a classification, yet upon firing, may cause that same classification to be retracted. RIMCRDR asserts that rules must be inferred in strict chronological order, and that no retractions are allowed [16]. In making these restrictions, it is felt that the RIMCRDR method will alienate some experts (since they will feel artificially restricted when the system does not allow them to use a classification as a condition in a rule simply because it is an exception), and make it unsuitable for many complex domains where retractions may be necessary.

Having identified these concerns, an attempt was made to develop a new multiple classification RDR (MCRDR) based method which would preserve all the essential benefits and strategies of the RDR method, while augmenting it with the ability to create rules which can use classifications as conditions. It was determined that it should do this while offering minimal restrictions as to when and where the expert may define these rules.

3 Method

In order to achieve the goals outlined above, the authors revised the existing MCRDR approach substantially. Changes were made to both the knowledge representation approach and the inference strategy. These changes in turn necessitated some revisions to the knowledge acquisition process, including provisions for the detection of cyclic rule definitions and an update to the cornerstone case mechanism. The resultant method has been entitled Multiple Classification Ripple Round Rules (MCRRR).

3.1 Knowledge Representation

The existing MCRDR knowledge representation, an n-tree, was deemed inadequate for the purposes of this new method. It was, however, desirable to maintain all the essential benefits that this structure offers. To this end, the structure was altered, becoming essentially a directed graph, similar in some ways to Gaines' Exception Directed Acyclic Graph [17]. Importantly, the underlying n-tree structure was still present, with each rule being a node with one or more conditions and a classification; however, nodes were given the added ability to store zero or more classifications, which must be satisfied in order for the rule to fire. These classifications could each be defined as classifications which either must be present, or must *not* be present in order for the rule to fire. These additions were termed switches, to reflect their simple mode of operation. Each switch represented a classification, and maintained a counter such that whenever a classification was added to the result set during the inference strategy, every switch concerning that classification was incremented. Conversely, if a classification was removed from the result set, the counter was decremented. It was important that these switches be a counter, rather than a Boolean, since it is entirely possible that a particular classification may be reached by more than one path, making it necessary to know how many instances of that classification are still present in the result set.

In order to know when and where updates were required during an inference process, a store of dependencies was also necessary. By maintaining a list of dependents and dependencies for each classification, it can be easily determined which nodes must be revisited and re-inferred whenever a classification was added or removed from the result set.

3.2 Inference

The typical MCRDR inference strategy is quite simple, being largely similar to that of a depth first search, where the deepest satisfied node is added to the result set, only where the search does not stop until every node at the first level of the tree has been traversed.

The new inference strategy must be substantially more complex, since there is now the possibility of nodes being revisited and of results being removed. There is also the additional need to check that all switches are active, before adding a rule, but this is a trivial step. The new inference algorithm for MCRRR is shown in a simplified pseudo-code form here.

```
infer(Node, Case) :-
{
        clearResult(Node, Case)
        If (Node's rule is satisfied
        AND all of its children's rules aren't)
                If (All Node's parent rules are satisfied)
                        Add Node to result list
                        Mark Node as having fired
                        Activate all dependents of Node's class
                                For each node that changed state
                                infer(ActivatedNode, Case)

        If (Node has a non-root parent rule)
                clearResult(Node's parent)
        For each Child that isn't marked as avoid
                Clear avoid markers
                infer(Child, Case)
}

clearResult(Node, Case) :-
{
        If (Node is marked as having fired)
                Remove it from the result set
                Clear its fired flag
                Deactivate all dependents of Nodes class
                For each node that changed state
                        infer(DeactivatedNode, Case)
                If (Node has a non-root parent rule
                AND all Node's parent rules are satisfied
                AND no siblings are satisfied)
                        Mark Node to avoid
                        infer(Node's parent, Case)
}
```

3.3 Knowledge Acquisition

From the user's perspective, the knowledge acquisition process remains largely unchanged from MCRDR. They are presented with a case, and the system's current "belief" regarding which classifications apply to it. If the expert believes a classification is missing, or that a classification is incorrectly provided by the system, they indicate as such and enter the rule creation process much as is done in MCRDR. During this process the expert may select what the classification should be and which valid conditions of the case are relevant to this classification, as normal. The only difference here is that the expert is also able to select any of the classifications the system is currently aware of as conditions of the case. If the classification is currently present on the case, then it can be added to reflect that the classification must be present in order for this rule to fire. If it is not currently present, the condition will be added to reflect that the classification must *not* be present in order for this rule to fire.

Cycles. Where the knowledge acquisition process does change is largely behind the scenes. Whenever the expert creates a rule, the system must check that their new rule does not have any potential to cause a cycle in the knowledge base. An example of a cycle can be seen in Figure 2, where each node is represented by 3 boxes, the top left being the switches which must be on for the rule to fire, the bottom left being the

conditions which must be satisfied, and the bottom right being the classification that the rule will add to the result set if it fires. It can be seen that in a case where X, Y and Z are all satisfied, the inference algorithm would add ClassA, which in turn would cause it to consider the next rule, adding ClassB, which in turn would add the ClassC rule. This ClassC rule is an exception to the rule which added ClassA, and thus would remove ClassA from the result set, which would in turn remove ClassB and so forth, with no termination possible.

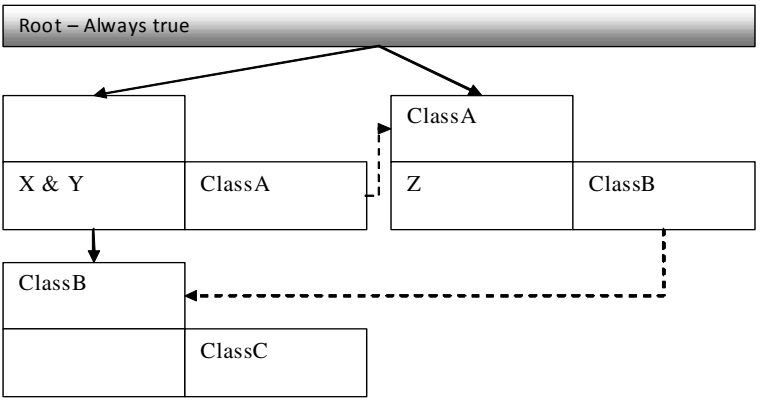


Fig. 2. A simple example of a cyclic rule set

Perhaps the most common method of detecting cycles in a directed graph structure is to perform a topological sort on the graph, as described by Kahn [18]. If a sorted topology cannot be found then there is a cycle. This method is efficient when considering whole graphs, however it was noted that in the context of this method it is only necessary to check the dependencies that are relevant to the new rule. As such, a method was used where the dependencies of all the classifications present in the new rule were examined in turn to see if they lead to a condition where the classification was no longer valid. That is, is there a dependency chain such that the classification might ultimately be dependent on itself being false? If a potential cycle was identified, the expert was informed that their new rule could cause conflicts, and was asked to revise the rule until no further conflicts were found. However, this approach is not expected to perform, in a computational sense, any better than Kahn's topological sort.

Cornerstone cases. It was also necessary to adjust the cornerstone case strategy in light of the alterations made to the method. In MCRDR, a cornerstone case can be robustly defined as any case which has been previously considered and approved by the expert, but which would be altered by the addition of the new rule that the expert is trying to create [4]. From the user's perspective, the cornerstone case process remains essentially unchanged here. When they attempt to add a rule to the knowledge base, the system will prompt them with a list of any past cases which would be altered by this new rule, and they must either select differences which will eliminate the

cornerstone cases, or they must confirm that the new rule should in fact alter the past case. However, behind the scenes, there is a loss of efficiency. MCRDR was able to interrogate cornerstone cases very efficiently, as a reference to each cornerstone case could be stored against the nodes they relate to. With MCRRR this approach may become impractical, since it would now be necessary to also store each classification of each case in addition to the attributes. This in itself would not be such a problem, until it is considered that the classifications of the past cases may also be changed by the addition of the new rule to the knowledge base. When the complexities of attempting to maintain this library of cornerstone cases were fully considered, it was deemed simpler to just re-infer all past cases with a temporary version of the knowledge base that included the new rule. Each case which had the new rule in its result set after inferencing would be considered a cornerstone case. This is likely to cause significant efficiency concerns as both the number of past cases and the size of the knowledge base grows, but it was felt that the majority of systems are sufficiently small to be managed with this approach. However, this hypothesis required testing.

3.4 Task

At this stage, the new method required testing and validation in several areas.

Obviously it was unknown whether the issue of cycles might become a limiting factor in the expert's use of the system, in a true configuration task. An evaluation of this was performed using a human expert and a blocks placement configuration task. However, this evaluation will not be described here, due to limited space.

Another unknown quantity was precisely how useful the additional features of MCRRR might be, compared to the existing MCRDR approach.

Additionally, it had been identified that there was a risk of the computational performance of the method becoming inadequate as the knowledge base and number of cases grew, particularly when adding rules, since, in its current form, it necessitates an inference of every past case in the system for each added rule. Each inference takes longer as progressively more rules are added to the system, and more inferences must be run as progressively more cases are assessed by the expert. However, it was unclear as to when these computational concerns might become apparent, or what effect the use of classifications as conditions or exceptions, or combinations thereof might also have.

To test these concerns two forms of simulated assessment were undertaken.

3.4.1 Simulated Experts and Grouping Rules

The first form follows closely with the simulated expert evaluation approach used initially by Compton to evaluate RDR [2], and Kang for MCRDR [4]. It has subsequently been used several times [2, 3]. Under this approach, four sets of simulated experts are trained using InductRDR [19] on an established single classification machine learning dataset, by randomizing the order in which cases are seen. Each simulated expert is then applied to the dataset incrementally on a case-by-case basis and interrogated to determine what rule conditions it used to reach its indicated classification. The expert is simulated at several levels of proficiency by using a certain portion

of the total number of conditions which are identified for each classification. For example, a "stupid" expert might use only 25% of the known conditions, thus making very non-specific rules which have a high chance of causing later errors, whilst a "clever" expert might use 75% of them, resulting in a low chance of the rule causing later errors. For this study, each expert had four incarnations, ones that used 25%, 50%, 75%, and 100% of the available rule conditions respectively.

However, there has been significant progress made in the field of multi-label machine learning methods since Kang's efforts were undertaken [1]. As such, the approach has been updated by employing the binary relevance classifier found in the Mulan [20] extension to Weka [21]. This allows for the creation of a genuine multi-class simulated InductRDR simulated expert, requiring only minor modifications to the technique.

The resultant MCRDR knowledge base could then be compressed into a crude MCRRR knowledge base, by searching for groups of conditions which are used several times over, and replacing them instead with a grouping rule. These "grouping" rules can be assessed for efficacy by considering the overall reduction in the number of conditions they provide. Unfortunately, this reduction is unlikely to be particularly significant. Consider the example of a grouping rule provided in Figure 3. This example indicates a situation where P,Q and X,Y are used to reach an intermediate class "ClassA", which can then in turn be used to reach a final class "Class B". The total number of conditions in this knowledge base is 6. If we instead represented this example without using the grouping rule, we would have two rules: "If P,Q,H then ClassB" and "If X,Y,H then ClassB", which also contains 6 conditions. To actually reduce the number of conditions in the knowledge base, the grouping rule must be used more than once, but even then its improvement on the overall number of conditions is only fractional. Despite this, it is felt that the reduction in conditions metric is a useful measure of how effective the MCRRR additions are for a given domain. It must simply be kept in mind that the MCRRR additions may have other, less measurable benefits, when used by a real human expert on a complex domain.

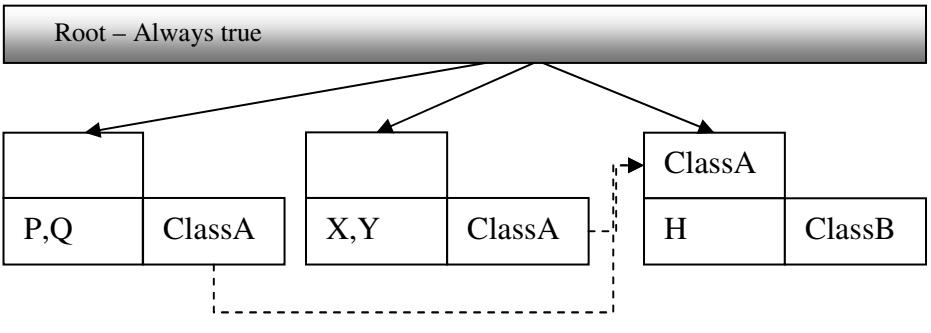


Fig. 3. A simple example of a (bad) grouping rule

The test was run for 7 freely available multi-label datasets, as shown in Table 1.

Table 1. The multi-label datasets used

		Attributes					
Name	**Domain**	**Instances**	**Nominal**	**Numeric**	**Labels**	**Cardinality**	**Density**
Emotions	Music	593	0	72	6	1.869	0.311
Genbase	Biology	662	1186	0	27	1.252	0.046
Scene	Multimedia	2407	0	294	6	1.074	0.179
Yeast	Biology	2417	0	103	14	4.237	0.303
Bibtex	Text	7395	1836	0	159	2.402	0.015
Enron	Text	1702	1001	0	53	3.378	0.064
Medical	Text	978	1449	0	45	1.245	0.028

3.4.2 Stress Test

The second method of simulated assessment undertaken was essentially a stress test, designed to test the computational performance of the method under a range of knowledge base environments. To do this, a rule generator was developed which would incrementally examine cases and create rules at random which satisfy the case. It would continue examining cases, creating one rule per case, until the limit (1000 rules/cases) was reached. Two adjustable parameters were used; the first indicated how likely each randomly generated rule was to be an *exception* to an existing rule (also selected at random), and the second indicated how likely they were to use a classification as a condition (the classification was also picked at random from the list of all classifications. The stress test was run for the enron (nominal) and scene (numeric) datasets, with a range of parameters, seen in Table 2. Each test was run 4 times.

Table 2. The parameters used in the stress test of each dataset

Rules	Cases	Exceptions	Classifications
1000	1000	10%	10%
1000	1000	20%	10%
1000	1000	40%	10%
1000	1000	10%	20%
1000	1000	20%	20%
1000	1000	40%	20%
1000	1000	10%	40%
1000	1000	20%	40%
1000	1000	40%	40%
1000	1000	10%	80%
1000	1000	20%	80%
1000	1000	40%	80%

It is anticipated that a performance in the rough order of $O(n^2)$ will be observed, reflecting an actual value of (cornerstone cases * rules), where cornerstone cases and rules are equivalent. However, it is also expected that the real time will increase in the simulations with higher intermediate rule chances, although not leaving order $O(n^2)$. In light of the fact that equivalent values of cornerstone cases and rules are being used, a 100 case inference was performed 9 times throughout each experiment to confirm that the actual time taken specifically to inference was progressing linearly with the number of rules in the system. The addition of this check allows for confirmation through elimination that the n^2 operation was a result of the combined effect of increasing numbers of cornerstone cases as well as increasing numbers of rules.

4 Results and Discussion

4.1 Simulated Experts and Grouping Rules

For each dataset, a table is shown to represent the average number of conditions in the knowledge base, both before and after conversion to MCRRR through the addition of grouping rules. The average reduction in conditions is also calculated.

Shown in Table 3 are the results found for each of the datasets tested. It can be seen here that the reduction in conditions was fairly minor, only 4.87% in the best case for the enron dataset. The emotions dataset also saw a modest reduction, despite having a limited number of rules to compress. The yeast, scene, bibtex, and medical datasets saw only minor reductions, while the genbase dataset simply did not have enough complex rules to see any reduction at all.

Table 3. The number of conditions in the various knowledge bases, before and after MCRRR conversion

	Avg. Conditions		
bibtex	**Before**	**After**	**Reduction**
25%	644	640.25	0.58%
50%	697.25	687.75	1.36%
75%	1244.25	1214.25	2.41%
100%	1467	1442.5	1.67%
emotions			
25%	56.3	55.5	1.42%
50%	52.7	51.9	1.52%
75%	84.3	81.7	3.08%
100%	160.3	154.2	3.81%

Table 3. (*continued*)

enron			
25%	956.6	918	4.04%
50%	925.7	880.6	4.87%
75%	1108.1	1059.7	4.37%
100%	1948.6	1860.1	4.54%
genbase			
25%	22.9	22.9	0.00%
50%	23.7	23.7	0.00%
75%	26.1	26.1	0.00%
100%	29.8	29.8	0.00%
medical			
25%	110.5	109.5	0.90%
50%	113.4	112	1.23%
75%	151.8	149.9	1.25%
100%	222.6	219.8	1.26%
scene			
25%	123.1	120.9	1.79%
50%	93.9	93.6	0.32%
75%	273.5	268.1	1.97%
100%	775.1	759.6	2.00%
yeast			
25%	122.1	120.6	1.23%
50%	160.5	157.2	2.06%
75%	563.6	550.4	2.34%
100%	1552	1520.4	2.04%

Of the datasets tested here, only the emotions and enron dataset appear to stand out as having any particular use for the grouping rules feature, with the enron dataset achieving respectable compression rates, and the emotions dataset achieving a surprising level of compression despite having limited rules to work with. Referring back to Table 1, unfortunately, no similarities can be detected between the properties of these two datasets. Of those datasets which performed relatively poorly, genbase, medical, scene, and bibitex, there also appears to be no identifying characteristics in Table 1.

As disappointing as these findings may be, they are not surprising. One feature all the datasets tested here share is that they were all designed for machine learning applications, and thus are all traditional machine learning problems. Certainly none of

these datasets are configuration or planning tasks, areas in which it is expected that the additional features of MCRRR might become required. Further to this, the metric itself is perhaps flawed, since an overall reduction in the number of conditions required to solve the problem may please experts because they potentially have less work to do to achieve the same result, but it is never expected to be the primary reason why an expert might want the ability to use classifications as conditions in their rule. One can imagine that the expert might want these features simply because expressing the rules in that manner "*makes more sense*" to them. So, although measuring condition reductions such has been done here might be easy, it appears, from these results at least, that it is not useful.

4.2 Stress Test

It was anticipated that a computational performance in the order of $O(n^2)$ would be observed, reflecting O(cornerstone cases * rules), where the number of cornerstone cases and the number of rules were roughly equal in quantity. This outcome was observed, for all parameters on both the enron and scene datasets. The effect of increasing the likelihood of exceptions had little to no impact on the outcome. However, increasing the likelihood of classifications as conditions did cause more sporadic outcomes, although never order of magnitude alterations. Examples of this can be seen in Figures 4 and 5.

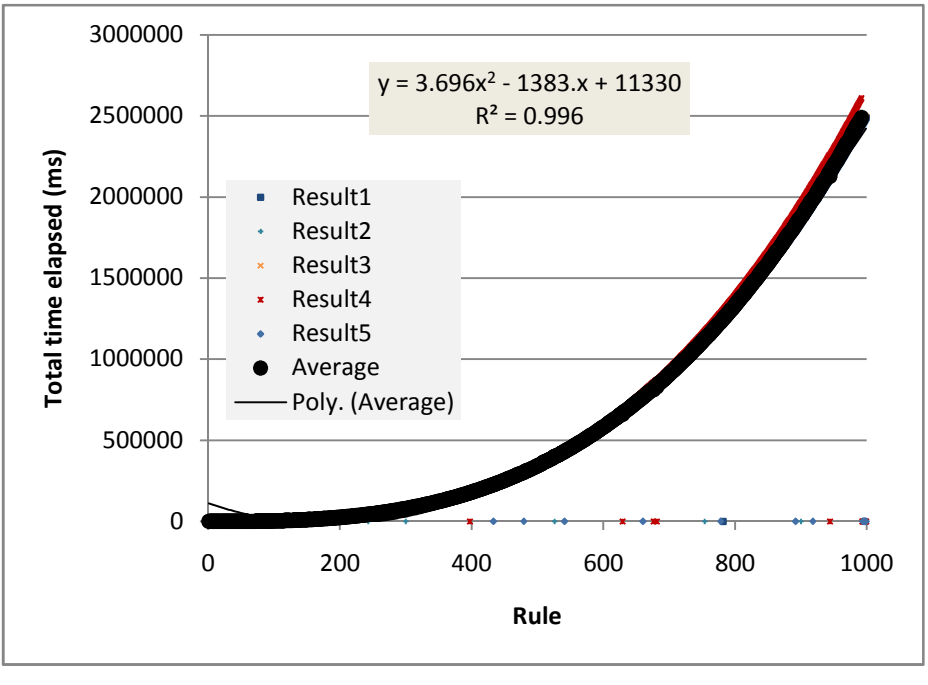


Fig. 4. A simulation with a 10% chance of exceptions and a 10% chance of rules using classifications

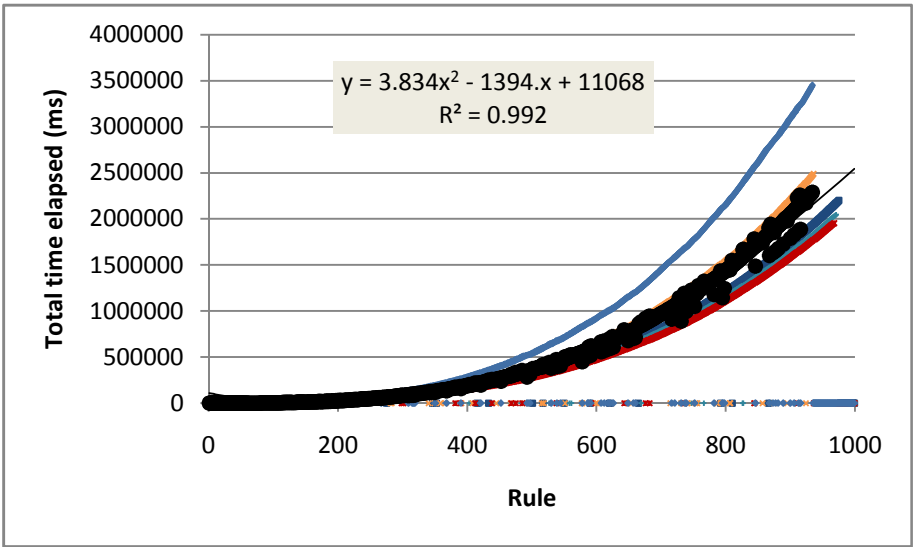


Fig. 5. A simulation with a 10% chance of exceptions and an 80% chance of rules using classifications

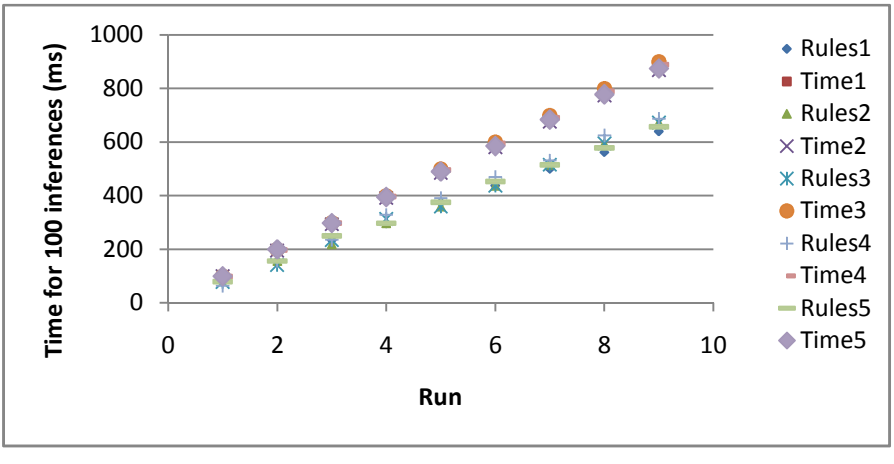


Fig. 6. The runtime of 100 inferences as the knowledge base is populated with gradually more rules

In order to demonstrate that the inference algorithm did perform linearly, proportional to the number of rules in the system, the performance of the inference algorithm alone was also measured. This was done by measuring the time it took to perform 100 consecutive inferences on a knowledge base at 9 points, after 100 rules were added, 200 rules were added etc. The result is shown in Figure 6, and indicates that the runtime of the inferences is directly and linearly proportional to the number of rules in the system, with the lower cluster of markers in each instance being the number of

rules in the system, and the higher cluster being the time taken to inference that knowledge base 100 times.

The unfortunate result of the performance measured here is that MCRRR in its current state should be considered unsuitable for domains where there is a very large number of cases or rules. On current hardware, the time taken to add a rule would become unacceptable after large numbers (>1500) of cases had been seen, and rules added.

5 Conclusions and Further Work

It has been determined that the MCRRR method is currently of limited value in many traditional domains, at least in the sense that the expert will find only limited use of grouping rules. However, it is still felt that the method may be of substantial value in other domains, such as configuration or planning tasks. However, proof of this remains to be published.

Unfortunately, it appears that the method as it currently stands is unsuitable in very large domains where many rules may be required, or many cases may be examined. However, it is thought that the computational performance of adding rules can yet be substantially improved. Currently, every cornerstone case must be inferenced for every rule added. However, through the appropriate application of indexing and searching strategies it should be possible to eliminate many of the past cases from the potential cornerstone case bank for any given rule, by simply examining the conditions of the rule. Of particular interest is the possibility of indexing cornerstone cases based on the values of their attributes, as well as the classifications they have. Having produced such an index it should be possible to substantially reduce the number of cornerstone cases which must be inferred upon, since many may be eliminated immediately due to not matching the search criteria of the new rule. This could potentially enable an order of magnitude performance increase, thus making the method applicable to even very large problem domains. It is felt that this is the important next step for MCRRR, and that with this problem resolved there will be no reason to justify using MCRDR rather than MCRRR, apart from the additional complexity of implementation.

References

1. Kang, B., Compton, P., Preston, P.: Multiple Classification Ripple Down Rules: Evaluation and Possibilities. In: AIII-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems, Banff (1995)
2. Compton, P., Preston, P., Kang, B.H.: The use of simulated experts in evaluating knowledge acquisition (1995)
3. Dazeley, R., Kang, B.H.: Rated MCRDR: Finding non-Linear Relationships Between Classifications in MCRDR (2003)
4. Kang, B.H.: Validating knowledge acquisition: multiple classification ripple-down rules. In: Computer Science and Engineering. University of New South Wales (1995)
5. Compton, P., Jansen, R.: A philosophical basis for knowledge acquisition. In: European Knowledge Acquisition for Knowledge-Based Systems, Paris (1989)

6. Compton, P., Kang, B.H., Preston, P., Mulholland, M.: Knowledge Acquisition without Analysis. In: Knowledge Acquisition for Knowledge-Based Systems. Springer, Heidelberg (1993)
7. Kang, B., Compton, P.: A Maintenance Approach to Case Based Reasoning. Advances in Case-Based Reasoning (1994)
8. Preston, P., Edwards, G., Compton, P.: A 2000 Rule Expert System Without a Knowledge Engineer. In: AIII-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems, Banff (1994)
9. Compton, P., Peters, L., Edwards, G., Lavers, T.G.: Experience with ripple-down rules. Knowledge Based Systems 19(5), 356–362 (2006)
10. Sarraf, Q., Ellis, G.: Business Rules in Retail: The Tesco.com Story, cited (2007)
11. Vazey, M., Richards, D.: Troubleshooting at the Call Centre: A Knowledge-based Approach (2005)
12. Bindoff, I., Tenni, P., Peterson, G., Kang, B.H., Jackson, S.: Development of an intelligent decision support system for medication review. J. Clin. Pharm. Ther. 32(1), 81–88 (2007)
13. Mulholland, M.: The Evaluation of the Applicability of Artificial Intelligence Software to Solving Problems in Ion Chromatography, in Chemistry. University of New South Wales (1995)
14. Beydoun, G., Hoffmann, A.: NRDR for the Acquisition of Search Knowledge. LNCS, pp. 177–186. Springer, Heidelberg (1997)
15. Compton, P., Richards, D.: Extending ripple down rules (1999)
16. Compton, P., Richards, D.: Generalising ripple-down rules. In: Knowledge Engineer-ing and Knowledge Management: Methods, Models, Tools, pp. 2–6 (2000)
17. Gaines, B.R.: Exception dags as knowledge structures
18. Kahn, A.B.: Topological sorting of large networks. Communications of the ACM 5(11), 558–562 (1962)
19. Gaines, B.R., Compton, P.J.: Induction of ripple down rules (1992)
20. Tsoumakas, G., Katakis, I.: Multi-Label Classification: An Overview. International Journal of Data Warehousing and Mining 3(3), 1–13 (2007)
21. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, San Francisco (2005)

The Ballarat Incremental Knowledge Engine

Richard Dazeley, Philip Warner, Scott Johnson, and Peter Vamplew

Graduate School of Information Technology and Mathematical Sciences, University of Ballarat, University Drive, Mount Helen, Victoria 3353, Australia
{r.dazeley,p.vamplew}@ballarat.edu.au, pjw@rhyme.com.au, scjohnno@gmail.com

Abstract. Ripple Down Rules (RDR) is a maturing collection of methodologies for the incremental development and maintenance of medium to large rule-based knowledge systems. While earlier knowledge based systems relied on extensive modeling and knowledge engineering, RDR instead takes a simple no-model approach that merges the development and maintenance stages. Over the last twenty years RDR has been significantly expanded and applied in numerous domains. Until now researchers have generally implemented their own version of the methodologies, while commercial implementations are not made available. This has resulted in much duplicated code and the advantages of RDR not being available to a wider audience. The aim of this project is to develop a comprehensive and extensible platform that supports current and future RDR technologies, thereby allowing researchers and developers access to the power and versatility of RDR. This paper is a report on the current status of the project and marks the first release of the software.

Keywords: Ripple Down Rules, Toolkit, Knowledge Based System, Machine Learning.

1 Introduction

Knowledge based systems (KBS) have become a common inclusion in many information processing systems. While early Expert Systems (ES) tended to be monolithic stand alone entities, the modern KBS tends to sit inside a larger system providing specialized functions for particular processes. This has allowed for an increase in the use of KBS technologies; however many of these are using traditional ES approaches that are not easily extended or maintained. These systems therefore have a limited life, which intern limits the life span of the system it is embedded.

The Ripple Down Rules (RDR) [1] family of methodologies have been widely recognized as a powerful production rule approach that addresses these issues. Research and development in RDR has been pursued for an extended period of time resulting in many refinements and commercial systems. However, researchers and developers have always developed their own implementations, which meet their direct need [2]. While some researchers have released their implementations for others to use, these are generally not easily extendable or applicable in other applications. Commercial implementations, as expected, have always been controlled by companies under strict licensing and are unavailable [2].

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 195–207, 2010.

This paper introduces an engine, referred to as the Ballarat Incremental Knowledge Engine (BIKE)[1] that is a comprehensive and extendable platform specifically designed for the RDR family of methodologies. The intention was that the engine will serve two primary purposes. The first was that the system should facilitate future research in RDR by being extendable and versatile. The second was for the system to provide a platform for developers to incorporate incrementally maintainable KBS solutions by including database integration services.

To accomplish these goals the engine was designed using a plugin architecture that allows any aspect of the system's behaviour to be overridden. For instance, changes to knowledge structure, the inferencing process or learning methodology can all be easily extended and modified. The following section will provide an overview of the various RDR approaches, followed by a brief discussion on implementations of RDR. Section 4 will discuss the first release of BIKE, focusing on the services it provides and aspects that allow it to be extended. Finally, we will discuss future extensions to the engine and discuss how researchers and developers can contribute to this project.

2 Overview of Ripple Down Rules (RDR)

Ripple Down Rules (RDR)[2] was first suggested by [1] as an approach to resolving the maintenance problem in knowledge based systems. It was argued that experts do not explain how a conclusion is reached; rather, that they justify their conclusion within a particular context [1, 2]. RDR was designed to capture this contextual information by storing the knowledge in an exception structure, where it was assumed that the context was the sequence of rules that were evaluated in reaching a conclusion [1, 3, 4]. This situation cognition view of knowledge [6, 7] resulted in the ability to capture expert's knowledge incrementally. Furthermore, the design of the methodology allows an expert system under development to validate knowledge without the need of a knowledge engineer or expensive testing procedures [4].

RDR uses a binary exception tree, where each node contains a rule, a conclusion, a cornerstone case and two branches: labeled as the 'true' (or 'exception') branch and a 'false' branch. During inferencing if a rule at the current node is found to be true then the true branch is followed and vice versa if the rule is false [1]. This process continues until a node is reached with no child down the appropriate branch. The conclusion returned is the one associated with the last successful rule.

For example (adapted from [7]), a case with the attributes {a, b, c, g, h} is presented to the RDR KB shown in Fig 1. In this tree it can be seen that: rules 1 and 3 have both true and false branches leading to further rules; rule 2 only has a false path; and, rule 6 (and the root node) only has a true path. When the case is presented it ripples down the tree using the path {0 – 1 – 3 – 6} where, because there is no attribute 'f' and no false branch, the inferencing process completes. The conclusion returned is 1 from rule 1, due to this being the last rule satisfied.

Learning in RDR is performed using a failure-driven approach [2] where the expert corrects a conclusion with which they disagree. After identifying a misclassification

[1] Documentation, software and source available at http://bike.ballarat.edu.au/

[2] RDR is sometimes referred to as Single Classification RDR (SCRDR).

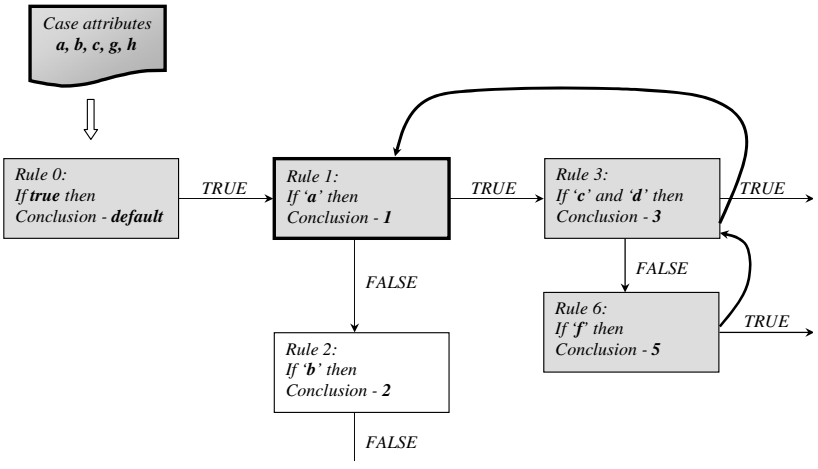


Fig. 1. Example of the RDR binary tree structure and inferencing process [7]

the expert simply provides the correct conclusion and a justification for why the original response was wrong. The justification is determined by the system first comparing the current case with the cornerstone case held at the node being corrected. A list of differences between these two cases is generated from which the expert selects one or more. The attributes selected by the expert are then used to justify the new rule. The new node includes a rule, the conclusion given by the expert and the case just processed as the new cornerstone case [1].

For example, Fig 2 illustrates how a new rule is created and added when the expert has decided the conclusion of class 1 is incorrect. Firstly, the cornerstone case from rule 6 is loaded and a difference list is extracted. The expert then selects the relevant differences that best distinguish between the documents, for instance 'h' and '!i' to form the new rule. A new node is then attached as child of rule 6 on the false branch containing the rule, the correct class given by the expert and our current case. The current case will become the cornerstone case for this new rule.

Since the creation of the original RDR there have been numerous extensions, modifications or new methodologies using the same philosophical basis. For instance, one popular adaption is Multiple Classification Ripple Down Rules (MCRDR) [8]. MCRDR is designed to be capable of producing multiple conclusions for each case, by using an n-ary tree where each child branch represents an exception to the parent [9]. Other approaches have been developed such as Nested RDR (NRDR) [10], Time Course RDR (TCRDR) [11], WISE [9] Dynamic RDR (DRDR) [12], Ripple down rule-Oriented Conceptual Hierarchies (ROCH) [13], MCRDR/FCA [14], Collaborative RDR (CRDR) [15] and Rated MCRDR [16, 17] to name just a few. The above approaches have been applied in a number of applications such as Labwizard by Pacific Knowledge Systems [18], KMAgent [2, 19], Knowledge Management Assistant (EMMA) [20] or embedded in other systems such as in a conversational agent [21] and planning [22]. RDR approaches have also been applied in machine learning through such techniques as InductRDR [23] and Cut95 [24]. For a more complete discussion readers are directed to the recent survey by [2].

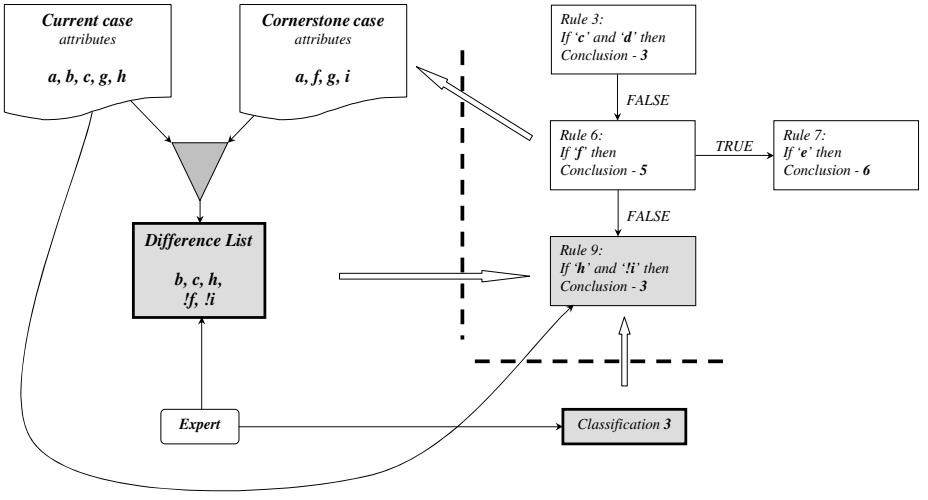


Fig. 2. Example of creating and incorporating new knowledge in RDR

3 Current Implementations

RDR methodologies have been implemented by several researchers and organizations over an extended period of time. The majority of researchers have developed their implementation from scratch to be used in their particular project [2]. Some of these researchers at various times have made these publicly available, such as Suryanto [2]. One of the most professionally developed RDR engines made publicly available was released by Associate Professor Byeong Kang and the MCRDR research group from the University of Tasmania in 2004[3]. This engine is a solid multiplatform implementation of MCRDR in Java and C. Unfortunately, these implementations are not easily extensible to new versions of the algorithm. Furthermore, they have not been updated or maintained since their initial release.

There have also been numerous commercial developments of the RDR engine. For instance the Pathology Expert Interpretative Reporting System (PEIRS) [25], Lab-Wizard developed by Pacific Knowledge Systems in 1996 [26], the Sonetto system developed by the Ivis Group[4], and the Yawl group that used RDR in their workflow management system [3]. The RDR engines at the core of these developments however have not been released for researchers or developers to utilize.

4 Ballarat Incremental Knowledge Engine

Work on BIKE started in 2008 to develop a simple RDR implementation to be used in current research and consultancies being performed by the University of Ballarat (UB) research staff. A partial implementation was first deployed in a decision support project for the Victorian Department of Justice in early 2009. Later in 2009 the

[3] http://www.appcomp.utas.edu.au/users/bhkang/
[4] http://www.ivisgroup.com/

project received funding through a Research Infrastructure Block Grant (RIBG) from UB to more extensively develop the system into a general engine. In 2010 the decision was made to release this engine publicly to allow researchers and students free access. The following sections will provide an extensive overview of the engines design and capabilities. It is not the aim of this paper to provide details of how to use or develop with the engine; such details will be made available on the engine's website http://bike.ballarat.edu.au/. This site includes a wiki based repository for researchers and developers to provide details of their work and articles published.

4.1 Overall Design

BIKE was designed from the ground up to be extendable. The intention was to be able to change any and all aspects of the system's operations through the use of plugins. The final design is sufficiently versatile that it could be used to implement many non-RDR based methodologies as well. Fig 3 provides a detailed overview of the system's architecture and identifies the central components of BIKE that have been implemented directly or via plugins. The system contains four primary components: the engine core, knowledge base, stream processing unit and a virtual expert. All of these components have been extended in a number of plugins. The plugins provided in the first release include a full implementation of both RDR and MCRDR, as well as extensions for both human and simulated experts. Other plugins have been included, such as the stream processor plugins and knowledge base plugins, as well as a simple user interface, which provide valuable support functionality.

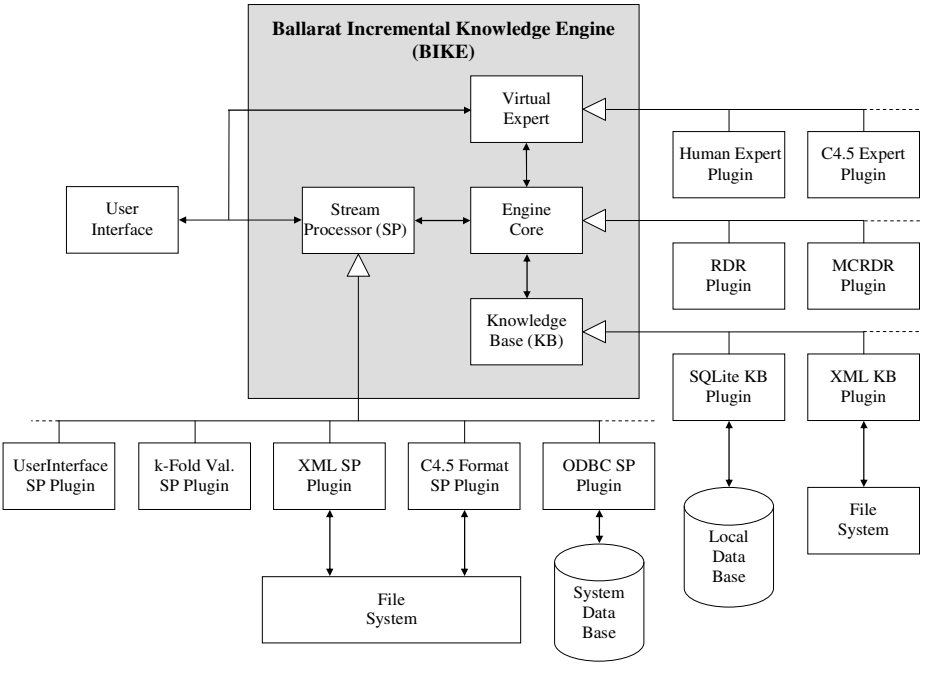


Fig. 3. Diagram illustrating BIKEs overall architecture

4.2 Engine Core

The engine core component provides the majority of the functionality for the system. The primary services provided by this component are the knowledge representation, plugin manager, expression manager, and the inference engine. It also provides a number of the general classes required for processing such as rules, cases, attributes, values, frames and results. The basic operation of this component is to receive a case from the stream processor to be processed by the nominated inference engine along with a knowledge base. The inference engine will process the case using its inference rules to guide its path through the knowledge representation and produce a result. This result is then passed back to a stream processor for post processing. Each of these elements will be described in more detail in the following subsections.

Knowledge Representation
Approaches to knowledge representation in RDR can be very diverse making the development of a generic system difficult. To get around this the core engine only implements a basic framework for how knowledge is represented. Plugins are then used to implement specific features of a particular methodology. The basic structure is based around an n-ary tree. Each `Node` in the tree can have any number of rules and any number of child nodes. Using this structure:

- In RDR the first child node can be regarded as an exception (true) branch and the rest of the children are the false nodes of the node before it, therefore, node *n+1* is the false child of node *n*. During inference RDR simply tests each child until one is found to be true, which is then followed. While this representation does not match the usual way it is conceived it does in fact match the original description of RDR [1].
- In MCRDR each child is simply regarded as an exception to the parent. Traversal is identical to RDR except instead of stopping when you find a child that evaluates to true you continue until all children have been tested, following each child that is true.
- A traditional KBS could also be represented by having a list of rules as a child of a place holding root node. It is then up to the inference engine to determine how the nodes will be traversed.

A planned addition for a future release is to also prove a `GraphNode` with a list of input and output arcs. Such a structure could be used to represent any graph like structure effectively allowing the addition of knowledge representations such as Collaborative RDR (CRDR) [15] or even concept graphs and semantic networks.

Inference Engine
The inference engine is the primary processing unit in the engine. It takes a case provided by an input stream and a knowledge base using the above representation scheme and is responsible for generating a result and sending it to an output stream. In BIKE the Inference Engine is represented with the `IAlgorithm` interface and by default does nothing. Plugins are responsible for implementing the inference engine functionality. However, while each plugin must implement their own version of the algorithm an array of services are provided for them to utilize, as discussed in the following subsections.

Attributes and Values

An `Attribute` is a simple class representing a name-value pair. A collection of attributes makes up a case and they are used in rules and difference lists. The attribute's name is a simply an identifier. There are a number of types of values that extend the interface `IValue`, represented in BIKE, which can also be defined within a plugin. The value types provided currently are:

- `BooleanValue`: A simple Boolean type.
- `IntegerValue`: Represents a value of type integer with up to 64 bits.
- `RealValue`: A value with double floating point precision
- `StringValue`: A value of type string using the Unicode character set.
- `ListValue`: A value that contains a list of other values.
- `Frame`: A Frame is a value that is based on a structured type in object oriented programming languages and will be discussed in the following subsection.

Each of these value types has a number of operations available. For instance `IntegerValue` and `RealValue` types have access to an array of operators: +, -, *, /, -, ^, comparison operators, min, max, sin, cos, tan, asin, acos, atan, atan2, round, DegToRad and RadToDeg. In these operations `IntegerValues` are promoted to `RealValues` where required. Likewise strings have comparison operators and concatenation, while lists have a collection of untyped meta-functions, such as ForAll(x in[...], <condition>), ForEach(...), ForAny(...), Count(...), and many more. Additionally, new `IValues` and associated operators can be easily created in a plugin which can be registered with the engine and used. For instance, a plugin could create a `ComplexValue` class and a `ComplexTypeDefinition` class for its operations. This new `IValue`, once registered, can be used with any algorithm.

Frames

One interesting type of `IValue` that has been included is the `Frame`. The frame value type allows the inclusion of structured types. A frame allows a value to be made up of a number of named `FrameSlots`, which in turn contain an `IValue`. Generally structured facts such as Frames are not used in RDR because it introduces issues in how to generate difference lists and how to build rules from such values. One exception to this is the work by [13] on ROCH which uses conceptual hierarchies. While none of the current plugins for BIKE use the frame facility, it was included to facilitate the potential development of systems like ROCH.

Cases

A `Case` is a class on which all inference engines operate. A case is essentially a collection of attributes that represent the state of the world for a given situation. In different situations a case may be used in various ways. For instance they may be loaded from a file either singly or in a batch process. Alternatively, the case can be created by loading data from a system database or by being entered by a user. The method the case is created however is unimportant to the engine core which simply takes a case via a stream processor (4.5) and applies the inference engine.

Rules and Expressions

A `Rule` is a `Persistable` class that maintains an association between an `Expression`, a `Conclusion` and, if required, a list of `CornerstoneCases`. While RDR only requires a single cornerstone case at each rule, MCRDR often may store multiple cornerstone cases at a node. The `Conclusion` class is also `Persistable` and contains an `IValue` representing the conclusion to be returned.

The `Expression` class is responsible for performing the evaluation of the rule. It represents an expression tree that contains any number of sub expressions. This structure exceeds the basic requirements of an RDR rule but allows for potential systems that require more advanced expressions. There are numerous types of expressions provided, which can be combined in any way required. The entire expression processing component of the engine is handled by the `ExpressionManager`. The `ExpressionManager` manages operator and function implementations, as well as the parsing of expressions. It stores the various types used by the evaluation process, which are registered with the `ExpressionManager` allowing any number of new types to be added. For instance, fuzzy logic based operations can easily be created and registered with the `ExpressionManager` allowing fuzzy expression resolution.

Results

A `Result` class is returned from each inference and contains details about what occurred. Primarily it contains two pieces of information: The path (sequence of rules) followed during inferencing, as well as the `Conclusion` found. In RDR each inference returns a `Result` with only one path and one `Conclusion`, while MCRDR will return a list of `Results` with corresponding `Conclusions`.

4.3 Knowledge Base

The Knowledge Base (KB) represents the storage layer for the engine. This component is responsible for ensuring all `persistable` objects are maintained in permanent storage. This component manages and maintains the knowledge base both on the permanent storage as well as in active memory. This design is particularly advantages to the development of large knowledge bases. The storage process is handled by the `IStorageManager` using interchangeable `StorageBackend` objects to store data and manage local in-memory caching. The storage manager knows about all `Persistable` objects through a smart `StoragePointer` that all objects created in the system use. These smart pointers also manage garbage collection allowing components not being used to be released dynamically.

The `IStorageBackend` interface has been extended in two separate plugins. The first of these provides the facility to write the KB in an XML file. The second stores the KB in a SQLite database. Also included is a general SQL backend that allows for a general ODBC backend layer, although the ODBC backend itself is not included in this release. An ODBC backend has the advantage that the knowledge base could be integrated with existing systems. Once again it is a relatively simple process to provide a plugin to control where and how a KB is to be stored.

4.4 Stream Processing

The Stream Processing (SP) component of BIKE provides and manages all processing operations and is key to providing BIKE's versatility. SPs all receive a stream of cases from some source, manipulate them in some fashion and then forward them on. The input and output of an SP can be another SP; therefore SPs can be piped together in numerous arrangements. Furthermore, everything in the BIKE processing life cycle is an SP. For instance the `Classifier` SP manages the entire process of operating the engine core. The `Classifier` SP accepts cases from some source (another SP) and passes it to the inference engine algorithm for classification. This design of attachable SP components can be a very powerful facility. For instance, it could be used for a propose and refine task by piping a series of `Classifier` objects in a series. There are three types of SPs available in BIKE: input, output and filters. BIKE requires at least one input, one output and ideally should have at least one filter but can have more. Fig 4 illustrates an overview of how the BIKE life cycle operates.

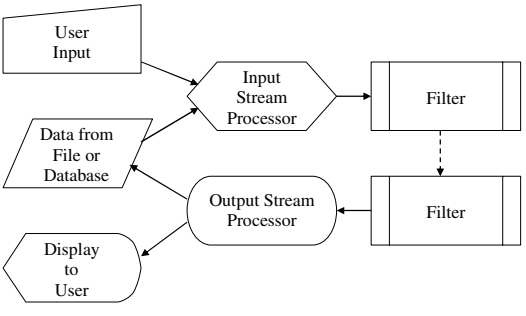


Fig. 4. Diagram showing the processing life cycle in BIKE. All systems must have an input and output stream processor and one or more filters, one of which will usually be a `Classifier`.

Input Stream Processors
Input SPs are responsible for gathering a stream of cases from a source such as a file, database or directly from a user. Like other components in BIKE a new SP can be provided in a plugin to add required functionality. Current input SPs are:

- `C45InputStream` – used to load cases using the C4.5 [27] file format.
- `XMLInputStream` – used to load cases using the XML file format.
- `ODBCInputStream` – used to load cases and data directly from a system database.
- `W32UserInputStream` – used to load cases and data directly from a Windows user interface.

Output Stream Processors
Likewise Output SPs are responsible for supplying a stream of cases to a file, database or directly to a user. Like an input SP a new output SP can be provided

in a plugin to add required functionality. Current output SPs provided with BIKE are:

- `C45OutputStream` – used to write cases to a C4.5 formatted file.
- `XMLOutputStream` – used to write cases to an XML formatted file.
- `W32UserOutputStream` – used to write cases to a Windows interface.

User Interfaces as Stream Processors
One interesting feature of BIKE is that there is no user interface directly implemented in the BIKE engine. Instead, a user interface in BIKE is developed as a type of input processing stream. In BIKE the input processing stream will first attempt to get the required details to form a case. A user interface SP simply gathers that case via the user interface and forwards it on to the next SP. One interesting feature of this is that SPs have the ability to send queries back to the SP that called them. Therefore, if an inference engine finds it is missing an attribute it requires, it can ask its calling SP. Each SP will then pass the request for the missing attribute back to its calling SP until one can resolve what the value should be. In a database SP this would be a query of the database, while in a user interface SP it will be a question displayed to the user. Likewise, a user output SP will display any response from the inference engine. The versatility of this approach is that we can attach any type of interface as an SP and it will automatically fit within the processing lifecycle of the engine. A second advantage is that any preprocessing of the attribute that is done, such as discretization, will be done automatically as the value is sent forward through the SPs. Currently the engine has implemented plugins for a console based input and output SP and a simple W32 based interface.

Filters
Filters are a little different in that they get and send their stream from and to another SP. Like other components a new filter can be provided in a plugin to add particular functionality required. Current filters provided with BIKE are:

- `Classifier` – Passes each case received to an inference engine for classification then forwards the combined case and result to the next SP. Allows the user to modify the knowledge base if the result is incorrect.
- `TestProcessor` – Passes each case received to an inference engine for classification then forwards the combined case and result to the next SP. Unlike the classifier it does not allow any corrections to be made to the knowledge base.
- `KFoldValidator` – This filter divides the input stream into equal sized pieces and forwards *k-1* to one filter and the k^{th} test fold to a second filter. It does this *k* times so that each fold has a turn at being the test fold. This is provided for general machine learning testing.

4.5 Virtual Expert

The Virtual Expert component with the base class `IExpert` provides a set of functions for getting a difference list from the inference engine and for creating rules through the selection of these differences. When a case has been processed by the inference engine the expert can indicate if the conclusion was incorrect. When this

occurs, the virtual expert will be asked to correct the error by selecting differences in the difference list. In various implementations the inference engine may also ask other questions of the virtual expert. The provision of the virtual expert allows developers to extend and modify the way the virtual expert responds.

Currently the virtual expert has been extended by three implementations: console expert, W32 expert, and C4.5 expert. The console and W32 experts link to user interfaces that allow a human to provide the expertise to respond to the requests by the inference engine. The C4.5 based expert is a simulated expert based on those used in a number of RDR papers [28, 29]. This expert is used during a simulated training session, where each misclassification by the engine is corrected by selecting attributes from a decision tree generated using C4.5.

4.6 Plugins

The engine has a `PluginManager` that loads all plugins located in the plugin folder upon startup. Plugins must be located in this folder with the appropriate system extension for the system to find it. Upon being loaded, each plugin registers itself and the various components it extends with the `PluginManager`, giving each a unique key string. Once a plugin is registered, a client program can request to use the facilities provided by the plugin by giving the matching key. Currently BIKE provides extension points for new stream processors (input, output streams and filters), algorithms, knowledge representation schemes, knowledge base storage, classification schemes, virtual experts, types, functions and operators.

5 Conclusion and Future Work

This paper has introduced the Ballarat Incremental Knowledge Engine (BIKE) and provided an overview of the design and functionality of the engine. The engine contains four core components: the engine core, knowledge base, stream processors and virtual expert. Also discussed is the multitude of plugins that have also been developed to extend the engines functionality. This development, however, is far from complete. It is expected that over time this engine can be continuously expanded and improved.

This project has now been released to researchers and developers to add additional plugins to extend its functionality. If the reader wishes to take part in adding their work to the BIKE platform they should visit the BIKE website at http://bike.ballarat.edu.au/[5]. Currently the future plans for the engine are to improve the functionality of the current W32 interface by adding visualization tools, as well as to create a web based interface for visitors to experiment with. We also plan to develop more of the commonly used RDR algorithms along with additional SPs.

The RDR approach has long been recognized as a powerful approach to developing maintainable knowledge based system, however until now its take up has been limited by the lack of an extendable and open implementation. This release of BIKE will have benefits both to researchers, as well as software developers. This will allow a much wider take up of the RDR family of methodologies and facilitate the further advancement of the branch of research.

[5] The website will be made available in August as the server is still being configured.

Acknowledgements

This project was funded by the University of Ballarat through a Research Infrastructure Block Grant (RIBG) in 2009.

References

1. Compton, P., Jansen, R.: Knowledge in Context: a strategy for expert system maintenance. Second Australian Joint Artificial Intelligence Conference (AI88) 1, 292–306 (1988)
2. Richards, D.: Two decades of Ripple Down Rules research. The Knowledge Engineering Review 24, 159–184 (2009)
3. Compton, P., Edwards, G., Kang, B., Lazarus, L., Malor, R., Menziès, T., Preston, P., Srinivasan, A., Sammut, C.: Ripple Down Rules: Possibilities and Limitations. In: 6th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW 1991). SRDG Publications, Canada (1991)
4. Compton, P., Kang, B., Preston, P., Mulholland, M.: Knowledge Acquisition without Analysis. In: Knowledge Acquisition for Knowledge Based Systems. Springer, Berlin (1993)
5. Menzies, T.: Towards Situated Knowledge Acquisition. International Journal of Human-Computer Studies 49, 867–893 (1998)
6. Dazeley, R., Kang, B.H.: Epistemological Approach to the Process of Practice. Journal of Minds and Machines, Springer Science+Business Media B.V. 18, 547–567 (2008)
7. Dazeley, R.: An Expert System Methodology for SMEs and NPOs. In: 11th Annual Australian Conference on Knowledge Management and Intelligent Decision Support - ACKMIDS 2008 (2008)
8. Kang, B.H., Compton, P.: Multiple Classification Ripple Down Rules. In: Third Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop. Japanese Society for Artificial Intelligence, Hatoyama, Japan (1994)
9. Kang, B.H.: Validating Knowledge Acquisition: Multiple Classification Ripple Down Rules. University of New South Wales, Sydney (1996)
10. Beydoun, G., Hoffmann, A.: NRDR for the Acquisition of Search Knowledge. In: Proceedings of Tenth Australian Joint Conference on Artificial Intelligence, Perth, Australia (1997)
11. Preston, P., Edwards, G., Compton, P., Litkouthi, D.: An Expert System Interpreter for Time Course Data with Refinement in Context. In: AAAI Spring Symposium: Artificial Intelligence in Medicine (1994)
12. Shiraz, G.M., Summut, C.A.: An incremental Method for Learning to Control Dynamic Systems. In: The Machine Learning Workshop of the IJCAI 1995, Montreal, Canada (1995)
13. Martinez-Bejar, R., Benjamins, V., Compton, P., Preston, P., Martin-Rubio, F.: A formal framework to build domain knowledge ontologies for ripple-down rules-based systems. In: 11th Banff Knowledge Acquisition for Knowledge Base System Workshop (KAW 1998), Canada, SRDG (1998)
14. Richards, D.: Ripple Down Rules with Formal Concept Analysis: A Comparison to Personal Construct Psychology. In: 11th Workshop on Knowledge Acquisition, Modeling and Management (KAW 1998), Banff, Canada, SRDG Publications, Department of Computer Science, University of Calgary, Calgary (1998)

15. Vazey, M., Richards, D.: Achieving rapid knowledge acquisition in a high-volume call centre. In: Kang, B., Hoffmann, A., Yamaguchi, T., Yeap, W. (eds.) Proceedings of the Pacific Knowledge Acquisition Workshop 2004, Auckland, pp. 74–86 (2004)
16. Dazeley, R., Kang, B.: Rated MCRDR: Finding non-Linear Relationships between Classifications in MCRDR. In: 3rd International Conference on Hybrid Intelligent Systems, pp. 499–508. IOS Press, Melbourne (2003)
17. Dazeley, R., Kang, B.H.: Generalising Symbolic Knowledge in Online Classification and Prediction. In: Richards, D., Kang, B.-H. (eds.) PKAW 2008. LNCS, vol. 5465, pp. 91–108. Springer, Heidelberg (2009)
18. Compton, P., Peters, L., Edwards, G., Lavers, T.: Experience with ripple-down rules. In: Proceedings of AI 2005, the Twenty-Fifth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, UK, pp. 109–121 (December 2005)
19. Park, S.S., Kim, Y.S., Kang, B.: Personalized Web Document Classification using MCRDR. In: Pacific Rim Knowledge Acquisition Workshop (PKAW 2004), Auckland, New Zealand. Springer, Heidelberg (2004)
20. Ho, V., Wobcke, W., Compton, P.: EMMA: an E-mail Management Assistant. In: Liu, J., Faltings, B., Zhong, N., Lu, R., Nishida, T. (eds.) IEEE/WIC International Conference on Intelligent Agent Technology, pp. 67–74. IEEE, Los Alamitos (2003)
21. Mak, P., Kang, B., Sammut, C., Kadous, W.: Knowledge acquisition module for conversational agents. In: Kang, B., Hoffmann, A., Yamaguchi, T., Yeap, W. (eds.) Proceedings of the Pacific Knowledge Acquisition Workshop PKAW 2004, Auckland, pp. 54–62 (2004)
22. Finlayson, A., Compton, P.: Incremental knowledge acquisition using RDR for soccer simulation. In: Kang, B., Hoffmann, A., Yamaguchi, T., Yeap, W. (eds.) Proceedings of the Pacific Knowledge Acquisition Workshop, PKAW 2004, Auckland, pp. 102–116 (2004)
23. Gaines, B.R., Compton, P.J.: Induction of Ripple Down Rules. In: Fifth Australian Conference on Artificial Intelligence (AI92). World Scientific, Hobart (1992)
24. Scheffer, T.: Algebraic Foundation and Improved Methods of Induction of Ripple Down Rules. In: Proceedings of the Pacific Knowledge Acquisition Workshop, PKAW 1996 (1996)
25. Edwards, G., Compton, P., Malor, R., Srinivasan, A., Lazarus, L.: Peirs: A pathologist-maintained expert system for the interpretation of chemical pathology reports. Pathology 25(1), 27–34 (1993)
26. Garsden, H., Basilakis, J., Celler, B., Huynh, K., Lovell, N.: A Home Health Monitoring System Including Intelligent Reporting and Alerts. In: EMBC 2004: Annual Conference of the Engineering in Medicine and Biology Society, San Francisco, CA (2004)
27. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo (1993)
28. Compton, P.: Simulating Expertise. In: Proceedings of the 6th Pacific Knowledge Acquisition Workshop, Sydney, Australia (2000)
29. Dazeley, R., Kang, B.: Detecting the Knowledge Boundary with Prudence Analysis. In: Wobcke, W., Zhang, M. (eds.) AI 2008. LNCS (LNAI), vol. 5360, pp. 482–488. Springer, Heidelberg (2008)

Finding Relation between PageRank and Voter Model

Takayasu Fushimi[1], Kazumi Saito[1], Masahiro Kimura[2], Hiroshi Motoda[3], and Kouzou Ohara[4]

[1] Graduate School of Administration and Informatics, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
{j09118,k-saito}@u-shizuoka-ken.ac.jp
[2] Department of Electronics and Informatics, Ryukoku University
Otsu, Shiga 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[3] Institute of Scientific and Industrial Research, Osaka University
Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp
[4] Department of Integrated Information Technology, Aoyama Gakuin University
Kanagawa 229-8558, Japan
ohara@it.aoyama.ac.jp

Abstract. Estimating influence of a node is an important problem in social network analyses. We address this problem in a particular class of model for opinion propagation in which a node adopts its opinion based on not only its direct neighbors but also the average opinion share over the whole network, which we call an extended Voter Model with uniform adoption (VM). We found a similarity of this model with the well known PageRank (PR) and explored the relationships between the two. Since the uniform adoption implies the random opinion adoption of all nodes in the network, it corresponds to the random surfer jump of PR. For an undirected network, both VM and PR give the same ranking score vector because the adjacency matrix is symmetric, but for a directed network, the score vector is different for both because the adjacency matrix is asymmetric. We investigated the effect of the uniform adoption probability on ranking and how the ranking correlation between VM and PR changes using four real world social networks. The results indicate that there is little correlation between VM and PR when the uniform adoption probability is small but the correlation becomes larger when both the uniform adoption and the random surfer jump probabilities become larger. We identified that the recommended value for the uniform adoption probability is to be around 0.25 to obtain a stable solution.

1 Introduction

Recent technological innovation in the web such as blogosphere and knowledge/media-sharing sites is remarkable, which makes it possible to form various kinds of large social networks, through which behaviors, ideas and opinions can spread. Thus, substantial attention has been directed to investigating the spread of influence in these networks [12,4,17]. The representative problem is the influence maximization problem, that is, the problem of finding a limited number of influential nodes that are effective for the

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 208–222, 2010.

spread of information through the network and new algorithmic approaches have been proposed under different model assumptions, e.g. descriptive probabilistic interaction models [5,15], and basic diffusion models such as independent cascade (IC) model and the linear threshold (LT) model [8,9,19]. This problem has good applications in sociology and "viral marketing" [1].

Another line of work on the spread of influence is opinion share analyses, i.e. how people changes their opinions, how each opinion propagates and what the final opinion share is, etc. A good model for opinion diffusion would be a voter model [13,16]. It is one of the most basic stochastic process model, and has the same key property with the linear threshold model that a node decision is influenced by its neighbors' decision, i.e. a person changes its opinion by the opinions of its neighbors. The basic voter model is defined on an undirected network with self-loop and each node initially holds one of K opinions, and adopts the opinion of a randomly chosen neighbor at each subsequent discrete time-step.

Even-Dar and Shapira [6] investigated the influence maximization problem (maximizing the spread of the opinion that supports a new technology) under the basic voter model with two ($K = 2$) opinions (one in favor of the new technology and the other against it) at a given target time T. They showed that the most natural heuristic solution, which picks the nodes in the network with the highest degree, is indeed the optimal solution, under the condition that all nodes have the same cost.

We propose a new model for the spread of opinions. Each person has a different influence on the other person and the person to person relation is directional. A person not only changes its opinion by its direct neighbors but also considers the overall opinion distributions of the whole society. The new model incorporates these factors and we call this model as an extended Voter Model with uniform adoption. Here we note that the new model has a strong similarity to the well known PageRank [2,11] which is an algorithm to rank Web pages. Since the uniform adoption can be viewed as random opinion adoption of all nodes in the network, it is equivalent to the random surfer jump of PageRank.

We mathematically derive the ranking vector of the new Voter Model and compare it with that of PageRank, and explore how the two models are related by a series of extensive experiments using four real world social networks. Especially we investigate the effects of the uniform adoption probability on node ranking and how the ranking of the new Voter Model and PageRank are correlated to each other with this probability. The ranking of the new Voter Model becomes the same as that of PageRank if we assume that the network is undirectional, but since both our new model and PageRank use directional network, the ranking results are not the same. The results indicate that the correlation varies with the uniform adoption probability. There is little correlation between the extended Voter Model and PageRank when the uniform adoption probability is small and the high ranked nodes are different, but the correlation becomes larger when both the uniform adoption and the random surfer jump probabilities become larger. We found that the ranking becomes stable for the uniform adoption probability in the range of 0.15 and 0.35 and the self correlation within the extended Voter Model is high in this region, which is consistent with the report that the recommended value for the random surfer jump is 0.15.

The paper is organized as follows. We briefly explain the standard Voter Model and revisit PageRank in sections 2 and 3, respectively. Then we explain our new Voter Model, the extended Voter Model with uniform adoption, in section 4. Experimental results that describe various correlation results are detailed in section 5. Finally we summarize our conclusion in section 7.

2 Voter Model

In this section, according to the work [6], we first consider the diffusion of opinions in a social network represented by an undirected (bidirectional) graph $G = (V, E)$ with self-loops. Here, V and E ($\subset V \times V$) are the sets of all the nodes and links in the network, respectively. For a node $v \in V$, let $\Gamma(v)$ denote the set of neighbors of v in G, that is, $\Gamma(v) = \{u \in V; (u, v) \in E\}$. Note that $v \in \Gamma(v)$.

According to the work [6], we recall the definition of the basic voter model with two opinions on network G. In the voter model, each node of G is endowed with two states; opinions 1 and 2. The opinions are initially assigned to all the nodes in G, and the evolution process unfolds in discrete time-steps $t = 1, 2, 3, \cdots$ as follows: At each time-step t, each node v picks a random neighbor u and adopts the opinion that u holds at time-step $t - 1$.

More formally, let $f_t : V \to \{1, 2\}$ denote the opinion distribution at time-step t, where $f_t(v)$ stands for the opinion of node v at time-step t. Then, $f_0 : V \to \{1, 2\}$ is the initial opinion distribution, and $f_t : V \to \{1, 2\}$ is inductively defined as follows: For any $v \in V$,

$$\begin{cases} f_t(v) = 1, \text{ with probability } \frac{n_{t-1}(1,v)}{n_{t-1}(1,v) + n_{t-1}(2,v)}, \\ f_t(v) = 2, \text{ with probability } \frac{n_{t-1}(2,v)}{n_{t-1}(1,v) + n_{t-1}(2,v)}, \end{cases}$$

where $n_t(k, v)$ is the number of v's neighbors that hold opinion k at time-step t for $k = 1, 2$.

3 PageRank Revisited

We revisit PageRank [2,11]. For a given Web network (directed graph), we identify each node with a unique integer from 1 to $|V|$. Then we can define the adjacency matrix $\boldsymbol{A} \in \{0, 1\}^{|V| \times |V|}$ by setting $a(u, v) = 1$ if $(u, v) \in E$; otherwise $a(u, v) = 0$. A node can be self-looped, in which case $a(u, u) = 1$. For each node $v \in V$, let $F(v)$ and $B(v)$ denote the set of child nodes of v and the set of parent nodes of v, respectively, $F(v) = \{w \in V; (v, w) \in E\}$, $B(v) = \{u \in V; (u, v) \in E\}$. Note that $v \in F(v)$ and $v \in B(v)$ for a node v with a self-loop.

Then we can consider the row-stochastic transition matrix $\boldsymbol{P}$, each element of which is defined by $p(u, v) = a(u, v)/|F(u)|$ if $|F(u)| > 0$; otherwise $p(u, v) = z(v)$, where $\boldsymbol{z}$ is some probability distribution over pages, i.e., $z(v) \geq 0$ and $\sum_{v \in V} z(v) = 1$. This model means that from dangling Web pages without out-links ($F(u) = \emptyset$), a random surfer jumps to page v with probability $z(v)$. The vector $\boldsymbol{z}$ is referred to as a personalized vector because we can define $\boldsymbol{z}$ according to user's preference.

Let $\boldsymbol{y}$ denote a vector representing PageRank scores over pages, where $y(v) \geq 0$ and $\sum_{v \in V} y(v) = 1$. Then using an iteration-step parameter t, PageRank vector $\boldsymbol{y}$ is defined as a limiting solution of the following iterative process,

$$\boldsymbol{y}_t^T = \boldsymbol{y}_{t-1}^T \left((1-\beta)\boldsymbol{P} + \beta \boldsymbol{e}\boldsymbol{z}^T \right) = (1-\beta)\boldsymbol{y}_{t-1}^T \boldsymbol{P} + \beta \boldsymbol{z}^T, \tag{1}$$

where $\boldsymbol{a}^T$ stands for a transposed vector of $\boldsymbol{a}$ and $\boldsymbol{e} = (1, \cdots, 1)^T$. In the Equation (1), β is referred to as the uniform jump probability. This model means that with the probability β, a random surfer also jumps to some page according to the probability distribution $\boldsymbol{z}$. The matrix $((1-\beta)\boldsymbol{P} + \beta \boldsymbol{e}\boldsymbol{z}^T)$ is referred to as a Google matrix. The standard PageRank method calculates its solution by directly iterating Equation (1), after initializing $\boldsymbol{y}_0$ adequately. One measure to evaluate its convergence is defined by

$$\|\boldsymbol{y}_t - \boldsymbol{y}_{t-1}\|_{L1} \equiv \sum_{v \in V} |y_t(v) - y_{t-1}(v)|. \tag{2}$$

Note that any initial vector $\boldsymbol{y}_0$ can give almost the same PageRank scores if it makes Equation (2) almost zero because the unique solution of Equation (1) is guaranteed.

4 Voter Model with Uniform Adoption

We propose an extended Voter Model with uniform adoption on a directed graph $G = (V, E)$ with self-loops for K opinions. Let $m_t(k, v)$ be the number of v's parents that hold opinion k at time-step t for $k = 1, 2, ..., K$. In addition, just like the personalized vector employed in PageRank, we introduce some probability distribution $\boldsymbol{z}$ over nodes. Let $m_t(k)$ be the weighted share of opinion k at time-step t given by

$$m_t(k) = \sum_{\{v \in V; f_t(v) = k\}} z(v), \tag{3}$$

then $f_t : V \to \{1, 2, ..., K\}$ is inductively defined as follows, given an initial opinion distribution $f_0 : V \to \{1, 2, ..., K\}$. For any $v \in V$,

$$f_t(v) = k, \quad \text{with probability } (1-\alpha)\frac{m_{t-1}(k, v)}{\sum_{k=1}^{k=K} m_{t-1}(k, v)} + \alpha m_{t-1}(k). \tag{4}$$

This model indicates that the opinion of each node $v \in V$ is influenced by its parents nodes $B(v)$ with probability $(1-\alpha)$ and by any other node $u \in V$ with probability α according to $\boldsymbol{z}$. Hereafter, α is referred to as the uniform adoption probability and the extended Voter Model with uniform adoption is referred to as VM for short[1].

Now we consider estimating the expected influence degree of node $u \in V$, which is defined as the expected number of nodes influenced by u's initial opinion $f_0(u)$. Note that the following definition does not depend on which opinion u holds initially. We denote the expected influence degree of node u at time-step t by $x_t(u)$. Let $\boldsymbol{h}_u \in \{0, 1\}^{|V|}$ be a vector whose u-th element is 1 and other elements are 0, and $\boldsymbol{Q}$ the column-stochastic

[1] We call it as the extended VM when we have to make distinction from the standard VM.

transition matrix, each element of which is defined by $q(u,v) = a(u,v)/|B(v)|$. Here note that $B(v) \neq \emptyset$ for any node $v \in V$ because of the existence of self-loop. From the definition of our model, we can calculate $x_1(u)$ as follows.

$$x_1(u) = (1-\alpha) \sum_{v \in F(u)} |B(v)|^{-1} + |V|\alpha z(u) = \boldsymbol{h}_u^T \left((1-\alpha)\boldsymbol{Q} + \alpha \boldsymbol{z}\boldsymbol{e}^T \right) \boldsymbol{e}. \tag{5}$$

Each element of the vector $\boldsymbol{h}_u^T((1-\alpha)\boldsymbol{Q} + \alpha \boldsymbol{z}\boldsymbol{e}^T)$ is the probability that the corresponding node v is influenced by the node u with one time-step. Thus from the independence property of the opinion diffusion process, we can calculate $x_t(u)$ as follows.

$$x_t(u) = \boldsymbol{h}_u^T \left((1-\alpha)\boldsymbol{Q} + \alpha \boldsymbol{z}\boldsymbol{e}^T \right)^t \boldsymbol{e}. \tag{6}$$

Here since the vector $\boldsymbol{h}_u$ works for selecting the u-th element, we can obtain the vector consisting of the expected influence degree at time-step t as follows:

$$\boldsymbol{x}_t = \left((1-\alpha)\boldsymbol{Q} + \alpha \boldsymbol{z}\boldsymbol{e}^T \right)^t \boldsymbol{e} = \left((1-\alpha)\boldsymbol{Q} + \alpha \boldsymbol{z}\boldsymbol{e}^T \right)^{t-1} \boldsymbol{x}_{t-1} \tag{7}$$

Moreover, since $((1-\alpha)\boldsymbol{Q} + \alpha \boldsymbol{z}\boldsymbol{e}^T)$ becomes the column-stochastic transition matrix, we can consider a stationary vector defined by $\boldsymbol{x} = \lim_{t \to \infty} \boldsymbol{x}_t$.

For the sake of technical convenience, we perform scaling to the vector $\boldsymbol{x}$ defined by $\boldsymbol{x} \leftarrow \boldsymbol{x}/|V|$. Then, similarly to PageRank calculation process defined in Equation (1), we can obtain the expected influence vector at time-step t as follows after initializing vector to $\boldsymbol{x}_0 = \boldsymbol{e}/|V|$:

$$\boldsymbol{x}_t = \left((1-\alpha)\boldsymbol{Q} + \alpha \boldsymbol{z}\boldsymbol{e}^T \right) \boldsymbol{x}_{t-1} = (1-\alpha)\boldsymbol{Q}\boldsymbol{x}_{t-1} + \alpha \boldsymbol{z}. \tag{8}$$

We can employ the same convergence measure defined by Equation (2), just by replacing the vector $\boldsymbol{y}$ with $\boldsymbol{x}$. Here, we note that in case of undirected networks with self-loops Equations (1) and (8) become completely equivalent since there exist no dangling nodes. Note also that in this case, our extended VM reduces to the standard VM by setting $\alpha = 0$. On the other hand, in case of directed networks with self-loops, Equations (1) and (8) give different vector sequences, and we empirically evaluate their differences with special emphasis on their stationary vectors.

5 Experiments

In this section, we evaluate the effects of 1) the uniform adoption probability α and 2) community structure, and examine the relation between VM and PR by extensive experiments using four real networks.

5.1 Experimental Settings

In our experiments, we employ the Pearson correlation coefficients as our basic evaluation measure. For the sake of convenience, we recall its definition: given two vectors, $\boldsymbol{x}$ and $\boldsymbol{y}$, the correlation coefficient $C(\boldsymbol{x}, \boldsymbol{y})$ is defined as follows.

$$C(\boldsymbol{x}, \boldsymbol{y}) = \frac{(\boldsymbol{x} - \bar{x}\boldsymbol{e})^T (\boldsymbol{y} - \bar{y}\boldsymbol{e})}{\sqrt{(\boldsymbol{x} - \bar{x}\boldsymbol{e})^T (\boldsymbol{x} - \bar{x}\boldsymbol{e})} \sqrt{(\boldsymbol{y} - \bar{y}\boldsymbol{e})^T (\boldsymbol{y} - \bar{y}\boldsymbol{e})}}, \tag{9}$$

where $\bar{x}$ and $\bar{y}$ stand for the average element values of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, and recall that $\boldsymbol{e}$ is a vector defined by $\boldsymbol{e} = (1, \cdots, 1)^T$.

As mentioned earlier, we focus on evaluating the vectors of the expected influence degree, each of which is the stationary vector defined as a limiting solution of Equation (8) in VM. In our experiments, the personalized vector $\boldsymbol{z}$ is set to uniform one, i.e., $\boldsymbol{z} = (1/|V|, \cdots, 1/|V|)^T$. Based on Equation (2), the convergence criterion to obtain the stationary vectors is set to $\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_{L1} < 10^{-12}$ in case of VM, and $\|\boldsymbol{y}_t - \boldsymbol{y}_{t-1}\|_{L1} < 10^{-12}$ in case of PR.

Our evaluation consists of two series of experiments. In the first series of experiments, we evaluate the effects of the uniform adoption probability on the expected influence degree. In the second series of experiments, we evaluate the effects of network's community structure on the expected influence degree. Now, we explain our method of rewiring the originally observed network to change its community structure. The rewired network is constructed just by randomly rewiring links of the original network according to some probability p without changing the degree of each node [14]. More specifically, by arbitrarily ordering all links except for self-loops in a given original network, we can prepare a link list $L_E = (e_1, \cdots, e_{|E|})$. Recall that each directed link consists of an ordered pair of *from*-part and *to*-part nodes, i.e., $e = (u, v)$. From the list L_E, we can produce two node lists, i.e., the *from*-part node list L_F and the *to*-part node list L_T. Thus, by swapping two elements of the node list L_T with the probability p so as not to produce multiple-links, we can obtain a partially reordered node list L'_T. Then, by concatenating L'_T with the other node list L_F, we can produce a link list for a rewired network. Namely, let L'_T be a shuffled node list, and we denote the i-th order element of a list L by $L(i)$; then the link list of the rewired network is $L'_E = ((L_F(1), L'_T(1)), \cdots, (L_F(|E|), L'_T(|E|)))$.

5.2 Network Data

In our experiments, we employed four sets of real networks, which exhibit many of the key features of social networks. Below we describe the details of these network data.

The first one is a reader network of "Ameba"[2] that is a Japanese blog service site. Blogs are personal on-line diaries managed by easy-to-use software packages, and have rapidly spread through the World Wide Web [7]. Each blog of "Ameba" can have the *reader list* that consists of the hyperlinks to the blogs of the reader bloggers. Here, a reader link from blog X to blog Y is generated when blog Y registers blog X as her favorite blog. Thus, a reader network can be regarded as a social network. We crawled the reader lists of 117,374 blogs of the Ameba blog service site in June 2006, and collected a large connected network. This network had 56,604 nodes and 1,071,080 directed links. We refer to this network as the Ameblo network.

Second one is a trackback network of blogs used in [9]. Bloggers discuss various topics by using trackbacks. Thus, a piece of information can propagate from one blogger to another blogger through a trackback. We exploited the blog "Theme salon of blogs" in the site "goo"[3], where a blogger can recruit trackbacks of other bloggers by registering an interesting theme. By tracing up to ten steps back in the trackbacks from the

[2] http://www.ameba.jp/

[3] http://blog.goo.ne.jp/usertheme/

blog of the theme "JR Fukuchiyama Line Derailment Collision", we collected a large connected trackback network in May, 2005. The resulting network had 12,047 nodes and 79,920 directed links. We refer to this network data as the Blog network.

The third one is a fan network of "@cosme"[4] that is a Japanese word-of-mouth communication site for cosmetics. Each user page of "@cosme" can have *fan links*. Here, a fan link from user X to user Y is generated when user Y registers user X as her favorite user. Thus, a fan network can be regarded as a social network. We traced up to ten steps in the fan links from a randomly chosen user in December 2009, and collected a large connected network[5]. This network had 45,024 nodes and 546,930 directed links. We refer to this network as the Cosme network.

Last we employed a network derived from the Enron Email Dataset [10]. We first extracted the email addresses that appeared in the Enron Email Dataset as senders and recipients. We regarded each email address as a node, and constructed a directed network obtained by linking two email addresses u and v if u sent an email to v Next, we extracted its maximal strongly connected component. We refer to this strongly connected bidirectional network as the Enron network. This network had 4,254 nodes and 44,314 directed links. We refer to this dataset as the Enron network dataset.

Table 1. Basic statistics of networks

network	$\|V\|$	$\|E\|$	$C(\boldsymbol{B}, \boldsymbol{F})$
Ameblo	56,604	1,071,080	0.61350
Blog	12,047	79,920	0.74377
Cosme	45,024	546,930	0.51940
Enron	19,654	377,612	0.54929

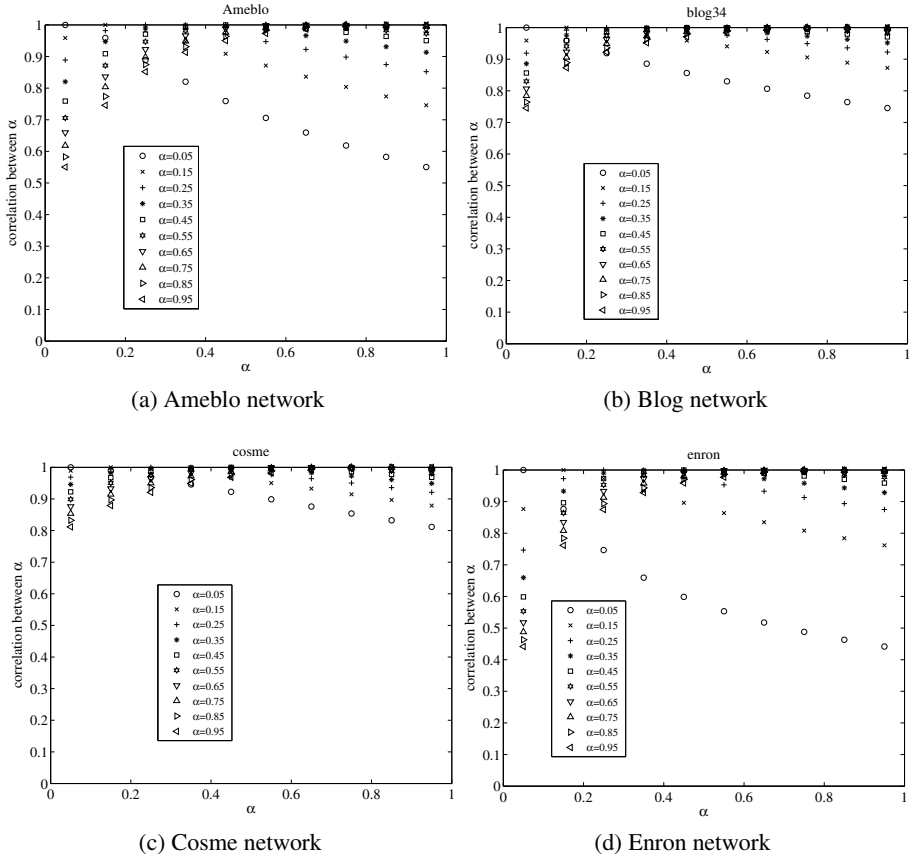
Table 1 shows the basic statistics of the Ameblo, Blog, Cosme and Enron networks. Here, $C(\boldsymbol{B}, \boldsymbol{F})$ denotes the Peason correlation coefficients between the in-degree vector $\boldsymbol{B}$, each element of which is $|B(v)|$, and the out-degree vector $\boldsymbol{F}$, each element of which is $|F(v)|$. From this table, we consider that each network has an intrinsic characteristics as a directed network because $C(\boldsymbol{B}, \boldsymbol{F})$ is reasonably smaller than 1.

5.3 Effects of Uniform Adoption Probability

As the first series of experiments, we evaluated the effects of the uniform adoption probability change on the expected influence degree. Here, let $\boldsymbol{x}(\alpha)$ be the stationary vector defined as a limiting solution of Equation (8) for VM with α. In order to evaluate the effects of different uniform adoption probabilities, we calculated the correlation coefficients $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha'))$ with respect to each pair of the uniform adoption probabilities, α and α' (self correlation). In Fig.1, we plot $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha'))$ with respect to α, where each result with different α' is shown by a different marker. Here we changed both the values of α and α' from 0.05 to 0.95 with an increment of 0.1.

[4] `http://www.cosme.net/`

[5] We further tried this collection procedure twice, and compared the resulting networks. Then, we found that they overlapped 99.5%.



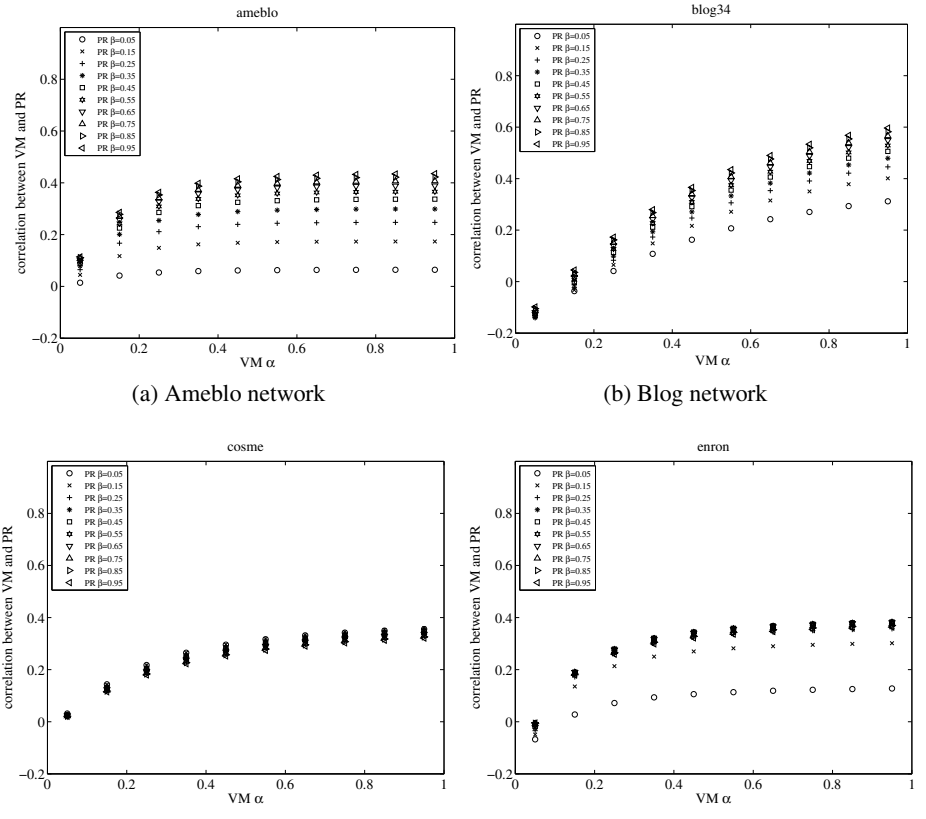
(a) Ameblo network (b) Blog network

(c) Cosme network (d) Enron network

Fig. 1. The correlation coefficient between VMs with different α

From Fig.1, we can observe the following smilar characteristics of VM for all of the four networks. First, the correlation coefficients $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha'))$ for any pair of α and α' are relatively high. Second, $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha'))$ in the range of $0.15 \leq \alpha \leq 0.35$ shows especially high values regardless of α. This suggests that we can recommend to employ this range of α because this would give a stable (and thus, representative) value of the expected influence degree for VM. Incidentally, it is reported that the uniform jump probability β in PR is frequently used at $\beta = 0.15$ [2,11]. Third, we can see that when $\alpha = 0.05$, $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha'))$ decreases almost linearly as α' increases, while it decreases very little for small values of α' and only modestly for large values of α' when $\alpha = 0.95$.

Similarly to the above, let $\boldsymbol{y}(\beta)$ be the stationary vector defined as a limmiting solution of Equation (1) for PR with β. In order to examine the relation between VM and PR, we calculated the correlation coefficients $C(\boldsymbol{x}(\alpha), \boldsymbol{y}(\beta))$ with respect to each pair of the uniform adoption probability α and the uniform jump probability β. In Fig.2, we plot $C(\boldsymbol{x}(\alpha), \boldsymbol{y}(\beta))$ with respect to α, where each result with different β is shown by a different marker. Here we also changed the values of β from 0.05 to 0.95 with an increment of 0.1.



(a) Ameblo network (b) Blog network

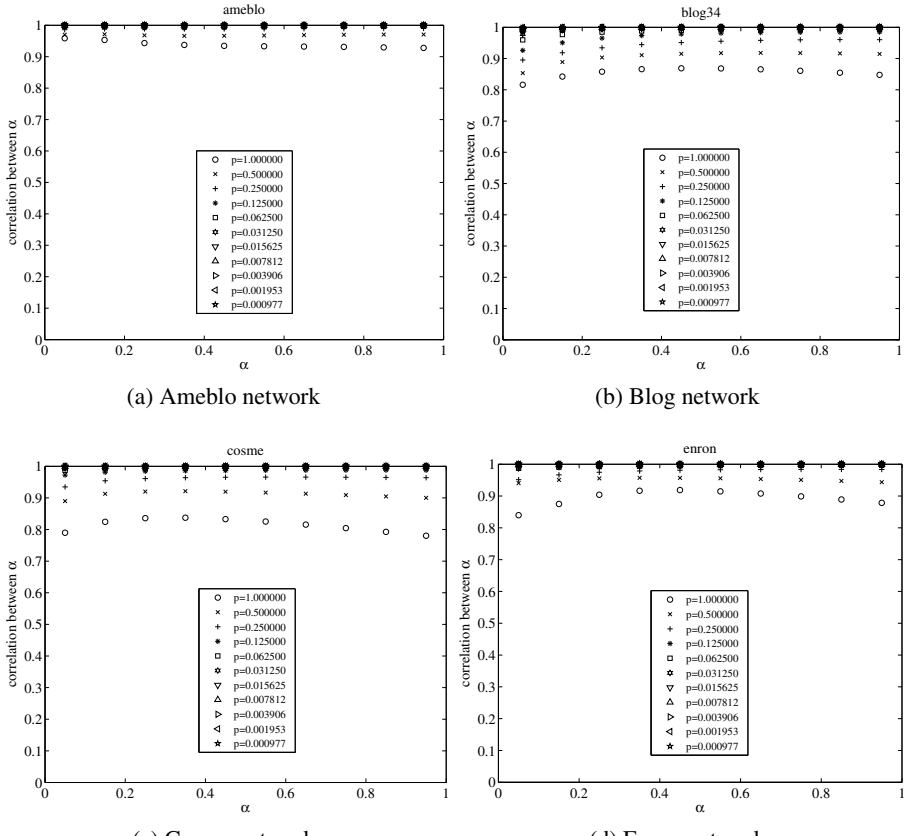
(c) Cosme network (d) Enron network

Fig. 2. The correlation coefficient between VM with α and PR with β

From Fig.2, we can observe the following similar relationships between VM and PR for all of the four networks. First, when α is small, there exists almost no correlation between the expected influence degree and the PageRank score. Second, for any β, $C(\boldsymbol{x}(\alpha), \boldsymbol{y}(\beta))$ generally increses as α increases, although their rates of increase depend on β as well as the network. Third, the maximum values of $C(\boldsymbol{x}(\alpha), \boldsymbol{y}(\beta))$ are attained at $\alpha = 0.95$. Incidentally, these maximum values are somewhat smaller than the correlation coefficients between in- and out-degree vectors, $C(\boldsymbol{B}, \boldsymbol{F})$, shown in Table 1, but their relative values are consistent between the two.

5.4 Effects of Community Structure

As the second series of experiments, we evaluated the effects of the community structure change on the expected influence degree. To this end, we constructed the 11 rewired networks from each of the original four networks using the rewiring probability $p = 2^{-k}$ ($k = 0, 1, \cdots, 10$) so that each network has a different community structure with different



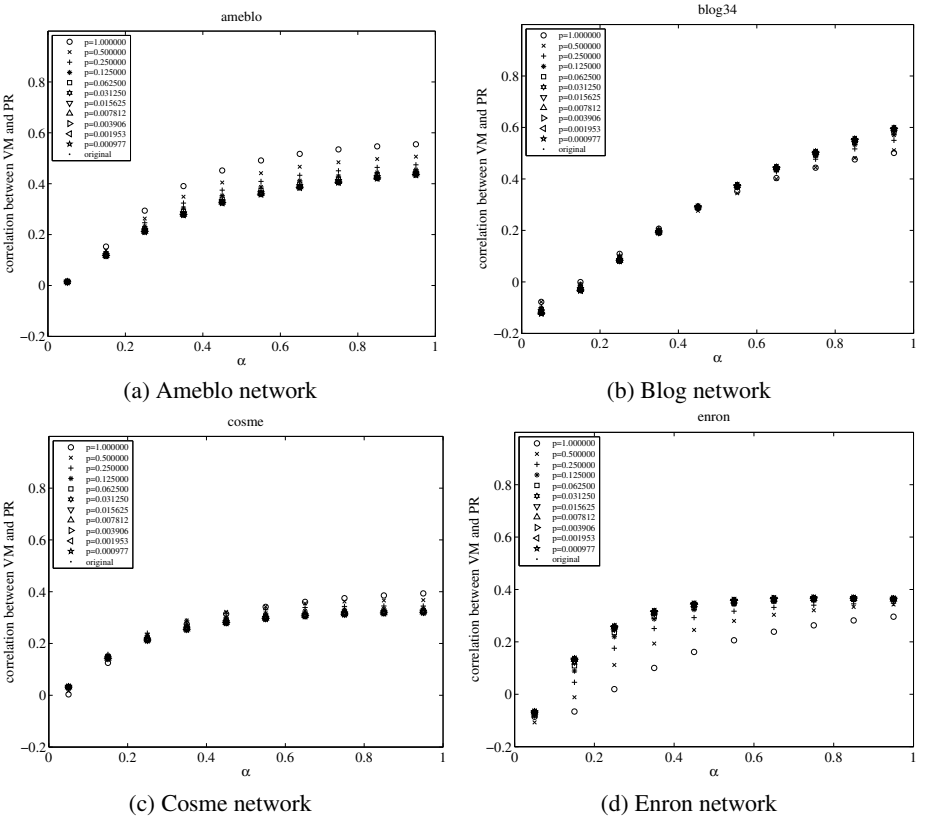
(a) Ameblo network (b) Blog network

(c) Cosme network (d) Enron network

Fig. 3. The correlation coefficient of VM between the original network and the rewiring network with p

degree from the original one's (see the rewiring method in Section 5.1). Now, let $\boldsymbol{x}(\alpha, p)$ be the stationary vector calculated from the network rewired with probability p for VM with α. In order to evaluate the effects of different community structure, we calculated the correlation coefficients $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha, p))$ with respect to each pair of the uniform adoption probability α and the rewiring probability p. In Fig.3, we plot $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha, p))$ with respect to α, where each result with different p is shown by a different marker. Again we changed the values of α from 0.05 to 0.95 with an increment of 0.1.

From Fig.3, we can observe the following smilar characteristics of VM for all of the four networks. First, the correlation coefficients $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha, p))$ for any pair of α and p are relatively high. Second, in comparison to Fig.1, there exist almost no ranges for α where $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha, p))$ gives especially high values for all values of p. Third, $C(\boldsymbol{x}(\alpha), \boldsymbol{x}(\alpha, p))$ monotonically decreases as p increases. Overall, this experimental results suggest that the expected influence degree is not much affected by the community structure although the effect is more for a network with less community structure.



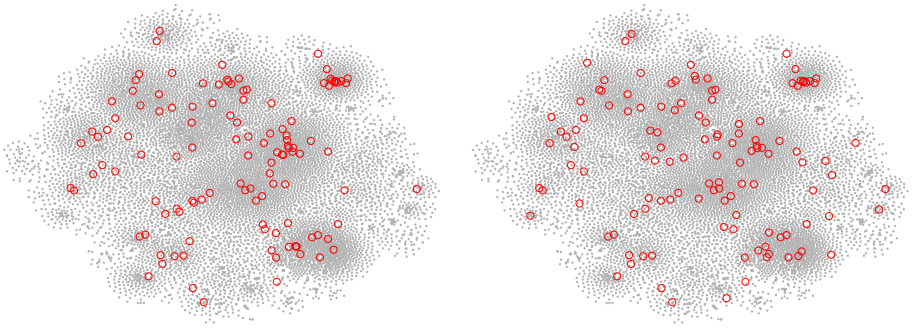
(a) Ameblo network (b) Blog network

(c) Cosme network (d) Enron network

Fig. 4. The correlation coefficient between VM with α and PR with $\beta(=\alpha)$ and p

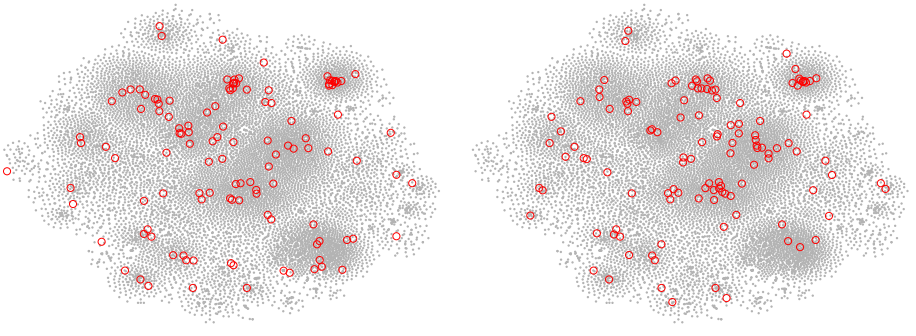
Similarly to the above, let $\boldsymbol{y}(\beta, p)$ be the stationary vector calculated from the network rewired with probability p for PR with β. In order to examine the relation between VM and PR in terms of community structure, we calculated the correlation coefficients $C(\boldsymbol{x}(\alpha), \boldsymbol{y}(\beta, p))$ with respect to each pair of the uniform adoption probability α and the rewiring probability p by setting $\beta = \alpha$. In Fig.4, we plot $C(\boldsymbol{x}(\alpha), \boldsymbol{y}(\alpha, p))$ with respect to α, where each result with different p is shown by a different marker.

From Fig.4, we can see that for all of the four networks, each plotting result is very similar to the corresponding one appearing in Fig.2. Namely, the correlation coefficients $C(\boldsymbol{x}(\alpha), \boldsymbol{y}(\alpha, p))$ for any pair of α and p are relatively small. Further, this experimental results also suggest that the expected influence degree is not much affected by the community structure. As an interesting distinction, $C(\boldsymbol{x}(\alpha), \boldsymbol{y}(\alpha, p))$ is large when p is large for the Ameblo and Cosme networks, but a reverse tendency can be observed for the Blog and Enron networks. Clarifying this reason is left for our future work.



(a) Blog network at $\alpha = 0.25$ (b) Blog network at $\alpha = 0.95$

Fig. 5. Visualization of Networks (VM)



(a) Blog network at $\beta = 0.25$ (b) Blog network at $\beta = 0.95$

Fig. 6. Visualization of Networks (PR)

5.5 Visual Analyses

We further analyzed the effects of the uniform adoption probability on the expected influence degree by visualizing the original networks. More specifically, we embeded the nodes in each network into a 2-dimensinal space by using the cross-entropy method [18], and plotted them as points. Then, we emphasized the highly influential nodes that have the expected influence degree within the top 1 % by using (red) circles. In the following experiments we only show the results using the Blog network as an example, but similar results were obtained for the other networks.

Fig.5 are the visualization results for two different values of α. Here we set α to 0.25 and 0.95 becauase they are considered to give the most and the least representative values for the expected influence degree as discussed in Section 5.3. From Fig.5, we can see that the highly influential nodes scatter around the entire the network for both α values. This partly explains the reason why the expected influence degree is not much affected by the community structure. This figure also shows that these two visualization results are close to each other.

We also analyzed the resullts of PR to see if there is any difference between VM and PR. Fig.6 are the visualization results for PR, and the (red) circles are again the highly ranked nodes that have the top 1 % PageRank score. Here we set β to 0.25 and 0.95, the same as α. From Fig.6, we can also see that the highly ranked nodes scatter around entire the network for both β values. Although we see that these nodes are different from the results of VM, but there is no clear difference bewteen the results of different β values.

6 Discussion

In this section, we discuss further extensions to our VM by employing majority rules and non-uniform adoption probability.

In our VM, with probability $(1 - \alpha)$, the opinion of each node $v \in V$ is influenced by choosing one of its parents nodes $B(v)$ and by any other node $u \in V$ with probability α according to $\boldsymbol{z}$. That is, our VM deals with all neighbor nodes equally by chosing one of neighbors at random. In a situation where a node decides its opinion considering the opinions of more than one neighbor, the q-voter model is one of the basic stochastic models [3]. In this model, the opinion of each node $v \in V$ is influenced by its chosen q parent nodes when their opinion is the same. Similarly, our VM can be further extended by adding some majority rules. We can also extend our VM with non-uniform adoption probability, that is, it might be natural to assume that not all friends or acquaintances have the same influence on a given node. To this end, we can introduce the weighted transition matrix $\boldsymbol{Q'}$ whose each element is defined by $q'(u, v) = w(u, v) / \sum_{u' \in V} w(u', v)$. Here, $w(u, v)$ is the weight over the link from a node $u \in V$ to a node $v \in V$ and $w(u, v) > 0$ if $a(u, v) = 1$; otherwise $w(u, v) = 0$. By using the column stochastic transition matrix $\boldsymbol{Q'}$, we can revise the Equation (8) as follows.

$$\boldsymbol{x}_t = \left((1 - \alpha)\boldsymbol{Q'} + \alpha \boldsymbol{z}\boldsymbol{e}^T\right)\boldsymbol{x}_{t-1} = (1 - \alpha)\boldsymbol{Q'}\boldsymbol{x}_{t-1} + \alpha \boldsymbol{z}. \tag{10}$$

In future, we plan to analyze these further extended models.

7 Conclusion

We addressed in this paper the problem of estimating the influential nodes in a social network, and focused on a particular class of information diffusion model, a model for opinion propagation. The popular model for opinion propagation is the Voter model in which the main assumption is that people change their opinion based on their direct neighbors, i.e. via local interaction. We extended this model to include the fact that people's opinion is also affected by the overall opinion distribution of the whole society. The new model is called the Voter Model with uniform adoption (the extended VM). It assumes that the network is directional because the people to people relation is directional.

The uniform adoption implies the random opinion adoption of all nodes in the network. We came to notice that this mechanism is the same as the random surfer jump

of the well known PageRank algorithm. This motivated us to investigate the relationship between the extended VM and PageRank. We mathematically derived the ranking vector of the extended VM and compared it with that of PageRank, and explored how the two models are related by a series of extensive experiments using four real world social networks. The both models assume a directed network and give different rankings because the adjacency matrix is asymmetric. However, if we assume an undirected network in which the adjacency matrix is symmetric, the both models become identical and should give the same ranking. We investigated the effects of the uniform adoption probability on node ranking and how the ranking of the extended VM and PageRank are correlated to each other with this probability. The results indicate that the correlation varies with the uniform adoptiopn probability. The correlation is very small when the uniform adoption probability is small, but it becomes larger when both the uniform adoption and the random surfer jump probabilities become larger. However, the visualization rerults do not indicate the clear difference of the rankings between the different values of the uniform adoption probability. We also investigated how the different community structure affects the correlation, but did not see the strong effects. We found that the ranking becomes stable for the uniform adoption probability in the range of 0.15 and 0.35 and the self correlation within the extended Voter Model is high in this region. It is interesting to note that the reported recommended value for the random surfer jump of PageRank is 0.15, which is similar to our finding for the uniform adoption probability.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

1. Agarwal, N., Liu, H.: Blogosphere: Research issues, tools, and applications. SIGKDD Explorations 10, 18–31 (2008)
2. Brin, S., Page, L.: The anatomy of a large scale hypertextual Web search engine. In: Proceedings of the Seventh International World Wide Web Conference, pp. 107–117 (1998)
3. Castellano, C., Munoz, M.A., Pastor-Satorras, R.: Nonlinear q-voter model. Physical Review E 80, 041129 (2009)
4. Crandall, D., Cosley, D., Huttenlocner, D., Kleinberg, J., Suri, S.: Feedback effects between similarity and sociai infiuence in online communities. In: Proceedings of KDD 2008, pp. 160–168 (2008)
5. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of KDD 2001, pp. 57–66 (2001)
6. Even-Dar, E., Shapria, A.: A note on maximizing the spread of influence in social networks. In: Deng, X., Graham, F.C. (eds.) WINE 2007. LNCS, vol. 4858, pp. 281–286. Springer, Heidelberg (2007)
7. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. In: Proceedings of the 13th International World Wide Web Conference, pp. 107–117 (2004)

8. Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146
9. Kimura, M., Saito, K., Nakano, R., Motoda, H.: Extracting influential nodes on a social network for information diffusion. Data Mining and Knowledge Discovery 20, 70–97 (2010)
10. Klimt, B., Yang, Y.: The enron corpus: A new dataset for email classification research. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 217–226. Springer, Heidelberg (2004)
11. Langville, A.N., Meyer, C.D.: Deeper inside PageRank. Internet Mathematics 1(3), 335–380 (2005)
12. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. ACM Transactions on the Web 1(5) (2007)
13. Liggett, T.M.: Stochastic interacting systems: contact, voter, and exclusion processes. Springer, New York (1999)
14. Newman, M.E.J.: The structure and function of complex networks. SIAM Review 45, 167–256 (2003)
15. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: Proceedings of KDD 2002, pp. 61–70 (2002)
16. Sood, V., Redner, S.: Voter model on heterogeneous graphs. Physical Review Letters 94, 178701 (2005)
17. Wu, F., Huberman, B.A.: How public opinion forms. In: Papadimitriou, C., Zhang, S. (eds.) WINE 2008. LNCS, vol. 5385, pp. 334–341. Springer, Heidelberg (2008)
18. Yamada, T., Saito, K., Ueda, N.: Cross-entropy directed embedding of network data. In: Proceedings of the 20th International Conference on Machine Learning, pp. 832–839 (2003)
19. Yang, S., Chen, W., Wang, Y.: Efficient influence maximization in social networks. In: Proceedings of KDD 2009, pp. 199–208 (2009)

Mobile Sync-application for Life Logging and High-Level Context Using Bayesian Network

Tae-min Jung, Young-Seol Lee, and Sung-Bae Cho

Dept. of Computer Science, Yonsei University
134 Shinchon-Dong, Seodaemoon-Gu, Seoul, 120-749, Korea
{realone,tiras}@sclab.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Abstract. Recently, the global mobile device market is growing up and the functionalities of the mobile devices expand continually. As the expansion of mobile device usage and functionality, it is possible to collect various kinds of personal information such as photographs, GPS coordinates, and multimedia contents. Some researchers have studied data management and transfer of personal information collected in mobile environment. Also, many applications are developed for managing and transferring mobile personal information to a personal computer. However, new technology for effective management and visualization as well as data transfer is necessary to overcome difficulties to manage and interpret the information. In this paper, we propose an application with effective interface for search, summarization and analysis about mobile personal information. It also provides statistical analysis and visualization of user's movement patterns on the map which uses NAVER map Open API (open map API provided in Korean website).

Keywords: Mobile Device, Bayesian networks, Mobile Data management, Life logs, Meta Data.

1 Introduction

In 2006, the global mobile device market reached one billion units. The market expanded more than 22.5 percent from the previous year. According to EIC, the market is expected to expand more than 1.6 billion units in 2010. Mobile phones with digital camera made an appearance in early 2000, and the market share of mobile-phone with camera rose to 59% in 2008 from 23% in 2004 in global markets. During the same period, the market share of MP3 phone increased from 9% to 61%. As the market is growing up, the functionalities of mobile devices expand continually.

As the expansion of mobile device usage and functionality, it is possible to gather life logs (photo, GPS log, log of multimedia usage, etc) on mobile device. There are many kinds of mobile devices such as cellular phone, PDA, smart phone, MP3. The devices help us memorize and manage our life, and the importance of life log increases for personal management.

These days, many researchers are studying life log collection and visualization using mobile devices [1–3], and it requires software for synchronization between PC and mobile devices. Most previous synchronization software had used to transfer PIMs data from mobile device to PC. But nowadays, it transfers many kinds of logs

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 223–234, 2010.

because mobile device has various multimedia data. Nokia provided software called "Nokia Photo," which saves photo and SMS visualizes them as ordered by time and allows us attach annotations and comments [4].

In this paper, we propose a novel system for log collection, log management, retrieval of special information, abstraction of life style, statistical analysis and visualization on a mobile device. This system visualizes various kinds of personal information on map with GPS coordinates. Also we show the feasibility of the proposed method with a prototype of synchronization software.

2 Background

Microsoft Research Center collected many logs, which are photos using SenseCam, GPS log, windows count in PC, date, time and personal information, in MyLifeBits project. They are stored in MS SQL Server database and visualized with several applications [1]. But they focused visualization of GPS and photos only and did not develop the propose protocol of data transfer from mobile devices to database server.

In University of Helsinki, Nokia 60 series are used to collect life logs (voice data, photo, battery usages, GPS, phone logs, SMS, and so on) from a smart phone. The system could send collected data to sever using HTTP and Bluetooth [2]. But this research only focused life log collection and do not try to visualize them.

University of Melbourne in Australia gathered life log from ten users using Nokia 7610 handset. These logs in mobile devices are synchronized with server. The users could observe them using Nokia Life Blog Software. The Nokia Life Blog show photo, SMS, and other context, but the system visualized SMS and Photo only. The visualization tool is provided for Nokia users [3].

Fig. 1. Examples of Data Transfer Applications: Microsoft ActivSync and Windows Mobile Device Center, Nokia Photo, and Samsung PC Studio

Many mobile device manufacturers deployed data transfer application. Microsoft developed ActiveSync and Windows Mobile Device Center [5]. These applications are to synchronize personal information in desktop PC and PDA which has Microsoft Windows Mobile OS. Nokia also released Nokia Photo that upload photos to web blog and manage photos, SMS, etc [4]. Samsung AnyCall, LG Cyon and Pentech also provided similar applications. Fig. 1 shows various data transfer application. These applications helped us transfer phone book, photos, multimedia files such as MP3 to PC. Most of them provide only simple data transfer function and photo tagging. It is difficult to use various life logs for recollection and management of events in real life.

This paper proposes novel system that transfers collected logs from mobile device to desktop PC. The system provides tools for visualization special log search tool

management tool, summarization tool and high-level context such as statistical information and event or emotion of person and. Fig. 2 show overview of system. This system visualizes them on PC application, and is developed on Microsoft .NET platform in Windows. We also provide visualization on map by using NAVER Map API (http://map.naver.com/).

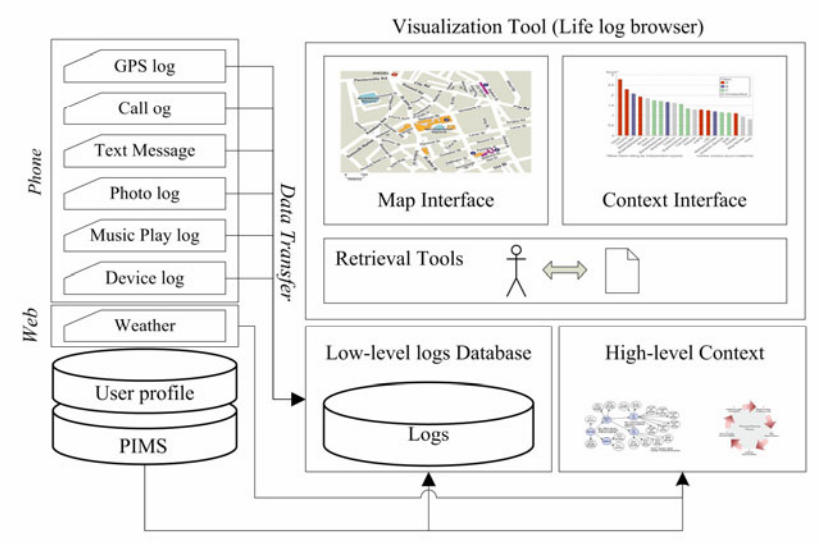


Fig. 2. Overview of Proposed System

3 High-Level Context

Here, we define high-level context as statistical information. The statistical information provides location, social network, call, and text message history, and extended definition includes semantic information that presents people's activity, special event or emotion [12]. We use Bayesian inference modules and statistics information analyzer in order to get this high-level context. Table 1 shows log information collected on a mobile device and on the internet. These logs are used for high-level context and visualization tools.

Table 1. The log information that was collected on a mobile device

Log	Information
GPS	Latitude, longitude, velocity, direction, date, time
Call History	Caller's phone number, type, time span, start/end time
Text Message	Sender's phone number, type, time span, start/end time
Photo	Photo file name, taking time
Weather	Weather, visibility range (km), cloud degree (%), temperature (℃), discomfort index, effective temperature (℃), rainfall (mm), snowfall (cm), humidity (%), wind direction, wind velocity (m/s), barometer (hPa)
MP3 Player	Title, time span, start/end time
Changing	Charging status, time span, start/end time

Bayesian network is used to infer context information, because it is an appropriate method to manage uncertainty (activity, and emotion in real life) in mobile environment [6, 8]. Bayesian networks for context inference are constructed by expertise (or expert knowledge). Mobile metadata are input to Bayesian network the input node of to infer context information [7, 9].

3.1 Bayesian Inference Modules

In order to take high-level context of activity, event or emotion of people, we used Bayesian networks. Various mobile log data is collected by mobile device, and then the Bayesian inference module detects the high-level context. The BN reasoning module performs probabilistic inference.

BNs refer to models that can express large probability distributions with relatively small costs to statistical mechanics. They have the structure of a directed acyclic graph (DAG) that represents the link (arc) relations of the node, and has conditional probability tables (CPTs) that are constrained by the DAG structure [11]. Fig. 3 shows an example BN that was designed by human and used for the application of this paper. It shows a DAG structure, node name, state name and inferred probabilities.

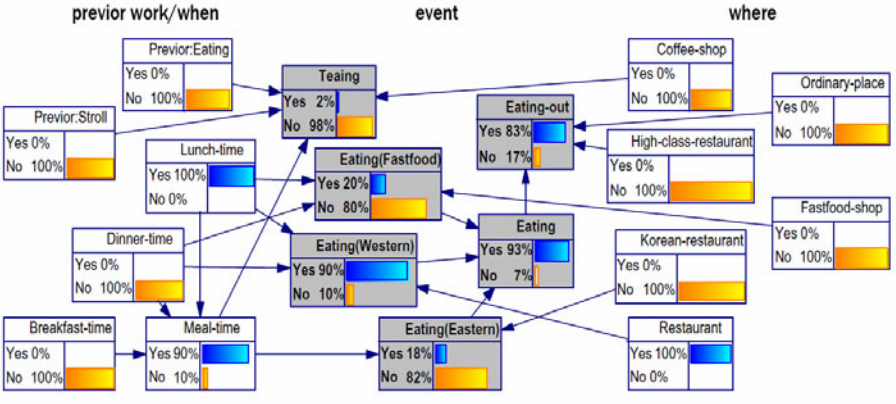


Fig. 3. The event inference BN designed for 'Activity in restaurant' in this paper

The high-level context inference is obtained by the belief probability, thresholds and weights of each Bayesian networks. The threshold is used to tune the context extraction model. The weights are used to apply the preferences or life-patterns of the given user, so if the context is a preferred by a user, the weight has higher value than others and the context can be selected more easily.

3.2 Statistical Information Analyzer

The other high-level context is statistical information. Statistics is used to analysis pattern of things. Statistics of life logs present people's life pattern. It can be also used to construct social network by call log and text message history. In this paper, we provide three-types of statistics information.

First is spatial statistics. It helps a user to know one's life pattern by frequently visited location and time. GPS data have just low level data, which it needs to translate semantic information. We use Location Positioning System (LPS) which is semiautomatic annotation tool. It provides map-based GUI, and allows a user to allocate personal semantics to specific spot. This system automatically updates location information to annotated (or registered) place. Finally statistical information analyzer used this semantic information for spatial statistics information by counting the frequency and time of visiting.

The statistics information for other log such as call history and text message has two standards. The one is person for construct social networking, and the other is day or weeks for life pattern. Call and text message are most important logs for social networking, because these logs has information about relationships among people using a mobile device. It can be measure of intimacy between a user and his friend. On the other hands, statistical information about day or weeks means degree of busyness. In person information extract by day and the others form whole data.

The statistical information is updated when new data are transferred from mobile device to PC, by statistical analysis. Our system visualizes daily and weekly statistics for SMS and call history in graphs, which can provide a user with interesting information. (It is mentioned chapter 4).

4 Visualization Tools

We propose system which data transfer application using life logging and high-level contexts. This system provide function data transferring mobile to desktop PC,

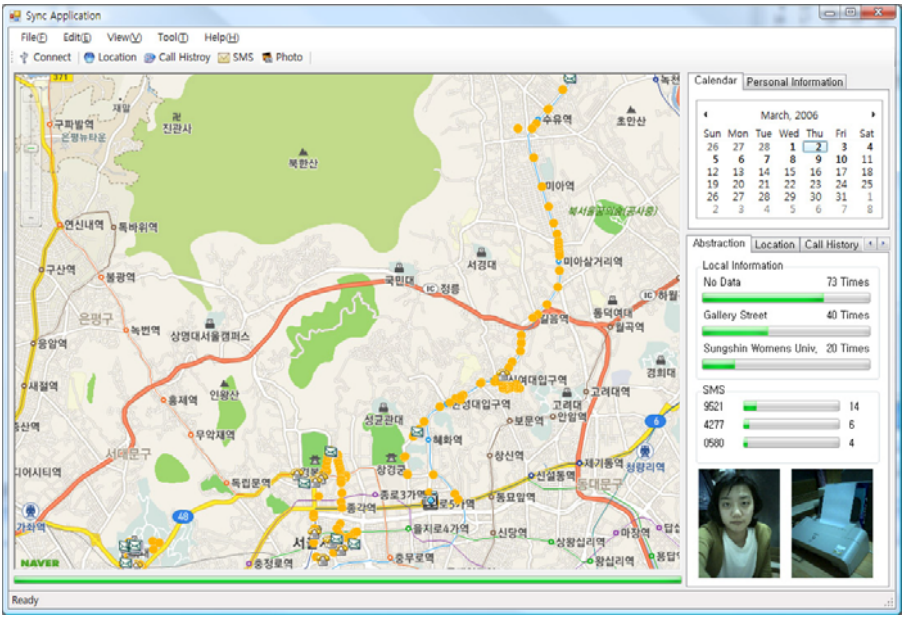


Fig. 4 A Screenshot of the Proposed System. It shows loaded one day life logs, the system provide visualization tool map, photo, SMS, call history, and Statistical Analysis.

and it can provide life logging service such as review one day, manage phone book, and so on.

The proposed system provides map-based interfaces. Spatial information well explains people's activity and life, because main interface in the proposed system is map. It can help low logs generated location and time. It supports location based search and browsing. And we proposed query-based search interface and statistical information visualization tools. Fig. 4 shows screenshot of proposed system.

4.1 Life Log Browser

Location information is meaningful in human life. As location affects people's activities and emotions, we provide browser tool using map interface for geographical presentation. People's locations are represented with orange circles on the map and call record, text message logs, photo logs are illustrated with icons. These icon have mouse over event, user can examine by just mouse moving. It allows users to explore various logs in short time. Fig. 5 explains how to show each log details.

Fig. 5. Log Visualization with Icons (A) Location information (GPS): place name, visited time, (B) Text Message: memo, type, send/receive time, sender call number, (C) Call record: send/receive time, type, sender call number, photo (if exist), (D) Photo log: place name, picture time

Table 2. Result of Limitation Distance Each Map Scale Level

Map Scale Level	1:200m	1:800m	1:3.2km	1:12.8km
Subject 1	0	75	90	201
Subject 2	0	51	84	254
Subject 3	0	30	72	222
Subject 4	0	99	114	241
Subject 5	0	48	78	222
Average	0	60.60	87.60	228.00

We define GPS sampling rate one/min, it enough abstract user's life. However, it has some problems. GPS logs may be lost because of battery problem or missed in the shadow of building or tunnel. Other logs (text message, photo, and so on) can be also observed while GPS signal is lost. In this case, all logs may be displayed on the same location of the map because many logs have the same information place. We use mouse click event for solving this problem. Clicking log view visualizes next logs in the same place.

Another problem is log centralization in life log browser. If too many logs are visualized on map interface, it is difficult to browse specific logs. The Higher map scale, the closer distance between logs is. We automatically change the number of log visualized on the map according to map scale. Euclidean distance between logs is calculated with GPS coordinate (latitude and longitude). We define a threshold of distance constraint empirically.

In order to get the best thresholds, we conduct experiments to determine thresholds of distance in each map scale by five people. Table 2 shows the result of the experiments, in which has four levels of map scale. Applied result is shown in Fig. 6.

This proposed life browser visualizes various logs on the map interface. It informs us of the locations where a user creates logs. It seems suitable for management

Fig. 6. Location Log Visualization with limitation distance. (A) is original view and (B) is view of limitation distance 87 (B). The map has the scale of 1: 3.2km.

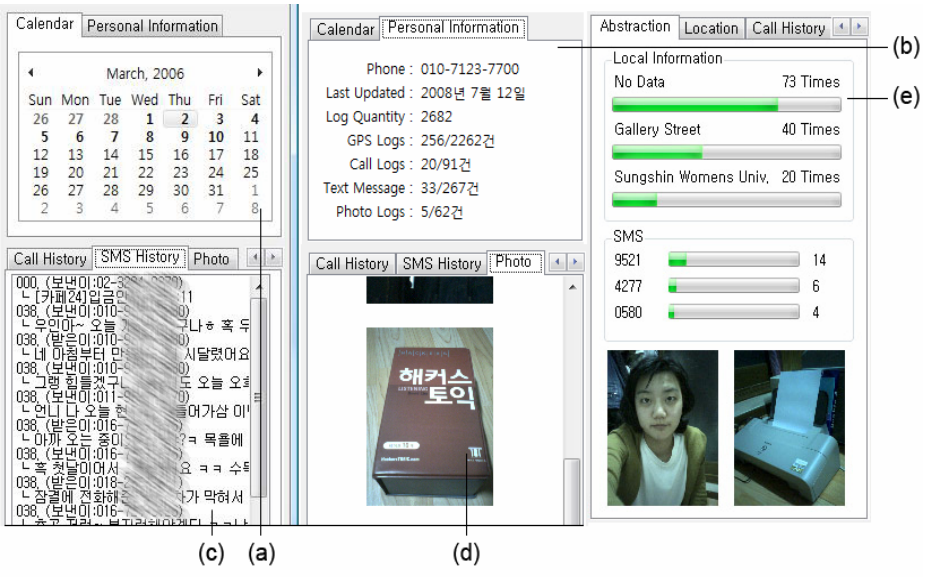


Fig. 7. (a) Calendar component: It helps to find some day. (b) Personal Information: It simple statistical logging information like total. (c) Text message history: It is ordered by time, and provides list view. (d) Photo logs: It is photo list and provide search by

human's life [13,14]. Additionally, it can use human memory assistance like memo or diary.

Map-based interface easily visualizes user's spatial patterns related to activity, but does not summarize personal temporal life patterns. We provide additional visualization tool for it. The interface has two components; one is a calendar component, and the other is list component. The calendar component is used to select and retrieve specific day or days from database for visualization. Second, list component shows the details of personal information sorted in temporal order. User can see the location of the item on map by double-clicking a specific item, which user wanted, in the list view.

4.2 High-Level Context Visualization Tools

The system provides two kinds of high-level context visualization tools. The one is graph interface for statistical information. It demonstrates various kinds of information. Fig. 8 is an example of this text message graph of visualization tool which provides pie, chart, curve, and other types of graph. The interface helps user understand life changes easily with resizable map and log list view. It can also interact with a user to find necessary logs on the map.

For visualization of inferred context on map, each GPS coordinate is assigned to each context. Various inferred context information can be placed at the same location. Each context has probability and priority [6, 9]. If context has

probability value more than 0.7 and priority more than 5, it is represented in bold characters. Change of context is marked as smile icons and shows the details by laying mouse over event. It can help human memorization and can be funny information [15].

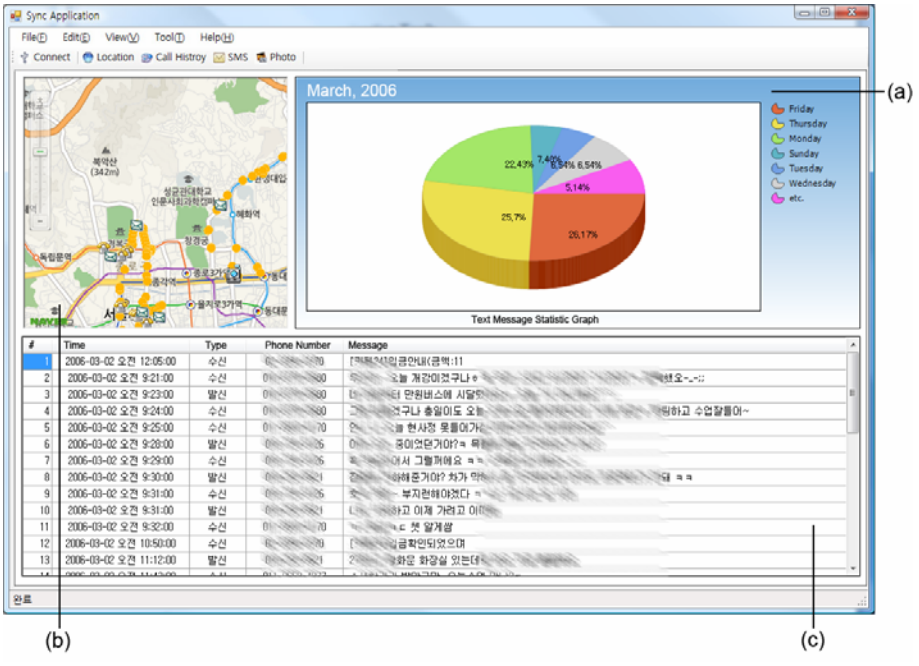


Fig. 8. An example of graph visualization (a) Graph interface, (b) Map interface, (c) List interface

5 Experiment and Result

Mobile personal data are collected in real life and the proposed method is applied to them to show the performance of visualization and summary. We also conduct usability test for evaluating the feasibility of the proposed system.

5.1 Apply to Real Data

Real logs are collected for three weeks by an undergraduate student who belongs to the department of computer science at Yonsei University. She took a bus for going to school. It reduces GPS logs error. We asked her to report events and schedule every day. Table 3 shows a schedule in a day, and screenshot of applied system is Fig. 4.

Table 3. An Example collected Logs in a Day

Time	*Place*	*Activity*	*Emotion*
7:30	Home	Get up and Ready for school	
8:00	Bus	Go to the school	Tired
9:30	University Lecture Room A	Major Class	
10:50	Library	Ready for Study	Busy
11:40	Bus	Move	
12:00	Samchung Dong	Lunch	Good
13:00	Bus	Move	
13:40	Bank	Date with boyfriend	
15:40	Bus	Move	Hastily
16:00	University Lecture Room A	Major Class	Funny
18:20	Bus	Move	Hastily
19:00	B University	TOEIC Study	Concentrate
21:00	Bus	Shopping and Move	Happy
21:30	Bus	Go Home	Happy

5.2 Usability Test

We conduct SUS (System Usability Scale) Test for objective evaluation of the proposed system. SUS test inspects three parameters: effectiveness (whether users seem successfully achieve their objectives or not), efficiency (how much effort and resource is required in achieving those objectives), satisfaction (whether the experience is satisfactory or not) [10]. Table 3 shows SUS questions. The scale of answers of each question ranges from 1 to 5 (strongly disagree to strongly agree). After that, we sum up all scores, multiply 2.5 and normalize the scores ranged between 0 and 100.

The test is conducted by 11 subjects. We show demo movie of the application and give it to them. Figure 10 shows the result of the SUS test. The average score is more than 72, confirming the reliable performance of this system.

Table 4. SUS (SYSTEM USABILITY SCALE) ITEMS

SUS	*Quesiton*
SUS-1	I think that I would like to use this svstem frequently
SUS-2	I found the system unnecessarilty complex
SUS-3	I thought the system was easy to use
SUS-4	I think that I would need the support of technical person to be able to use this system
SUS-5	I found the various function in this system were well integrated
SUS-6	I thought there was too much inconsistency in this system
SUS-7	I would imagine that most people would learn to use this system quickly
SUS-8	I found the system very cumbersome to use
SUS-9	I felt very confident using this system
SUS-10	I needed to learn a lot of things before I could get going out with this system

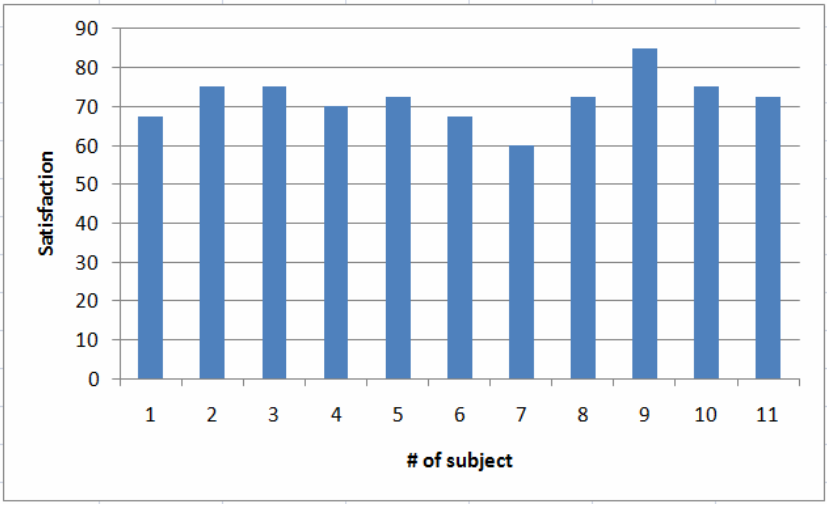


Fig. 9. Result of SUS test and average is 72

6 Conclusion

We develop a novel application for data transfer and visualization, and we also inference high-level context using Bayesian network. The system has various interfaces for life log visualization, browsing, retrieval, context view system. It will help memorization of human life and know life one's style. The usability of proposal system is evaluated through the SUS test and real data.

In future work, we will compare current logs and previous logs in real time. It helps user to find some changes of his or her life. It can be also used to recommend user adaptive service. We will implementation on mobile device. It provides real time service and can be assistant tool more concentrate human life.

Acknowledgement. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2009-0083838).

References

[1] Gemmell, J., Bell, G., Lueder, R.: MyLifeBits: Personal database for everything. Communications of ACM 49(1), 88–95 (2006)

[2] Raento, M., Oulasvirta, A., Petit, R., Toivonen, H.: ContextPhone: A prototyping platform for context-aware mobile applications. IEEE Pervasive Computing 4, 51–59 (2005)

[3] Hartnell-Young, E., Vetere, F.: Lifeblog: A new concept in mobile learning? In: Proc. of the IEEE Int. Workshop on Wireless and Mobile Technologies in Education, pp. 1–5 (2005)

[4] Nokia Photo, http://europe.nokia.com/get-support-and-software/download-software/nokia-photos
[5] Micrsoft Windows Mobile, http://www.microsoft.com/windowsmobile/
[6] Hwang, K.-S., Cho, S.-B.: Modular Bayesian networks for inferring landmarks on mobile daily life. In: The 19th Australian Joint Conference on Artificial Intelligence, pp. 929–933 (2006)
[7] Korb, K.B., Nicholson, A.E.: Bayesian Artificial Intelligence. Chapman & Hall/CRC (2003)
[8] Horvitz, E., Dumais, S., Koch, P.: Learning predictive models of memory landmarks. In: CogSci 2004: 26th Annual Meeting of the Cognitive Science Society, pp. 1–6 (2004)
[9] Hwang, K.-S., Cho, S.-B.: Life log management based on machine learning technique. In: Proc. of IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008), pp. 691–696 (2008)
[10] Brooke, J.: SUS: A "quick and dirty" Usability Scale, Usability Evaluation in INdustry. Taylor and Francis, London (1996)
[11] Korb, K.B., Nicholson, A.E.: Bayesian artificial intelligence. Chapman & Hall/ CRC (2003)
[12] Dourish, P.: What we talk about when we talk about context. Personal and Ubiquitous Computing 8(1), 19–30 (2004)
[13] Korpipaa, P., Mantyjarvi, J., Kela, J., Keranen, H., Malm, E.-J.: Managing context information in mobile devices. IEEE Pervasive Computing 2(3), 42–51 (2003)
[14] Krause, A., Smailagic, A., Siewiorek, D.P.: Context-aware mobile computing: Learning context-dependent personal preferences from a wearable sensor array. IEEE Transactions on Mobile Computing 5(2), 113–127 (2006)
[15] Rhee, Y.-H., Kim, J.-H., Chung, A.: Your Phone Automatically Caches Your Life. ACM Interactions, 42–44 (July-August 2006)

Consensus Clustering and Supervised Classification for Profiling Phishing Emails in Internet Commerce Security

Richard Dazeley[1], John L. Yearwood[1], Byeong H. Kang[2], and Andrei V. Kelarev[1]

[1] Centre for Informatics and Applied Optimization
Graduate School of ITMS, University of Ballarat
P.O. Box 663, Ballarat, Victoria 3353, Australia
{r.dazeley,j.yearwood,a.kelarev}@ballarat.edu.au
[2] School of Computing and Information Systems
University of Tasmania, Private Bag 100
Hobart, Tasmania 7001, Australia
BHKang@utas.edu.au

Abstract. This article investigates internet commerce security applications of a novel combined method, which uses unsupervised consensus clustering algorithms in combination with supervised classification methods. First, a variety of independent clustering algorithms are applied to a randomized sample of data. Second, several consensus functions and sophisticated algorithms are used to combine these independent clusterings into one final consensus clustering. Third, the consensus clustering of the randomized sample is used as a training set to train several fast supervised classification algorithms. Finally, these fast classification algorithms are used to classify the whole large data set. One of the advantages of this approach is in its ability to facilitate the inclusion of contributions from domain experts in order to adjust the training set created by consensus clustering. We apply this approach to profiling phishing emails selected from a very large data set supplied by the industry partners of the Centre for Informatics and Applied Optimization. Our experiments compare the performance of several classification algorithms incorporated in this scheme.

1 Introduction

The applications of clustering techniques to profiling phishing emails and web sites is an important problem in internet commerce security, which has been actively investigated recently. Various clustering algorithms have been used in this context by many authors. To illustrate, here we refer to just a few recent articles on this topic [7,15,16,18].

This paper investigates a novel combined method, which at the same time uses unsupervised consensus clustering algorithms as well as supervised classification algorithms. First, a variety of independent clustering algorithms are

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 235–246, 2010.

applied to a randomized sample of the data. Second, several consensus functions and sophisticated algorithms are used to combine these independent clusterings into one consensus clustering. Third, the consensus clustering of the randomized sample is used as a training set to train several fast classification algorithms on the consensus clustering of the initial sample. Finally, these fast supervised classification algorithms are used to classify the whole large data set in order to obtain final clustering.

This approach makes it possible to apply slow and most reliable clustering methods at the initial stages to improve the accuracy. It increases the speed of processing the whole large data set by incorporating fast algorithms in the final stages. It facilitates the inclusion of contributions from the domain experts to adjust the initial training set created by consensus clustering algorithms.

Our experimental investigation applies this approach to profiling phishing emails selected from a very large data set supplied by the industry partners of the Centre for Informatics and Applied Optimization. Our experimental results compare the efficiency of performance of several classification algorithms incorporated in this approach.

The number k of clusters is chosen and fixed as an input parameter for our algorithms. The question of how to vary this number and choose the most appropriate one for any given application is not considered in the present article.

The outcomes obtained show that this method can be used as a novel way of combining several clustering techniques in order to classify very large data sets of phishing emails for subsequent forensic analysis based on the resulting individual clusters.

The paper is organised as follows. Section 2 is devoted to the preprocessing of data and extraction of features for clustering algorithms. Section 3 outlines unsupervised clustering algorithms applied to obtain an initial clustering ensemble for a small randomized sample of the data set. Section 4 describes consensus functions and heuristics used to combine the ensemble into one final consensus clustering. Section 5 deals with the supervised classification algorithms trained on the consensus clustering. Section 6 summarizes the experimental results comparing the efficiencies of several classification algorithms in this setting.

2 Feature Extraction

Many authors have concentrated on the applications of machine learning algorithms for classification and clustering of phishing emails, since phishing represents one of the most rapidly growing and changing areas of internet commerce security. Phishing usually involves acts of social engineering attempting to extract confidential details by sending emails with false explanations urging users to provide private information that will be used for identity theft. The users may be requested to reply to the email, or visit a bogus web site, where they are asked to enter personal details, such as credit card numbers, tax file numbers, bank account numbers and passwords. More comprehensive information concerning phishing is presented, for example, by the Anti-Phishing Working Group [2] and OECD Task Force on Spam [19].

We have undertaken experimental investigation of this novel approach to clustering, outlined in Section 1, for a sample of 3276 emails randomly selected from a very large data set of phishing messages supplied by the industry partners of the Centre for Informatics and Applied Optimization. A flexible preprocessing and feature extraction system has been implemented in Python for the purposes of this investigation. It has been used to extract features concerning the content and structure of the emails, and hyperlinks embedded in the text.

Following [12], we used the *term frequency–inverse document frequency* word weights, or TF-IDF weights, as features for the clustering. These weights are defined using the following concepts and notation. Suppose that we are extracting features from a data set E, which consists of $|E|$ messages. For a word w and a message m, let $\mathrm{N}(w, m)$ be the number of times w occurs in m. Suppose that a collection $T = \{t_1, \ldots, t_k\}$ of terms $t_1, \ldots, t_k$ is being looked at. The *term frequency* of a word $w \in T$ in a message m is denoted by $\mathrm{TF}(w, m)$ and is defined as the number of times w occurs in m, normalized over the number of occurrencies of all terms in m:

$$\mathrm{TF}(w, m) = \frac{\mathrm{N}(w, m)}{\sum_{i=1}^{k} N(t_i, m)} \tag{1}$$

The *document frequency* of the word w is denoted by $\mathrm{DF}(w)$ and is defined as the number of messages in the given data set where the word w occurs at least once. The *inverse document frequency* is used to measure the significance of each term. It is denoted by $\mathrm{IDF}(w)$ and is defined by the following formula

$$\mathrm{IDF}(w) = \log\left(\frac{|E|}{\mathrm{DF}(w)}\right). \tag{2}$$

The *term frequency–inverse document frequency* of a word w in message m, or TF-IDF weight of w in m is defined by

$$\text{TF-IDF}(w, m) = \mathrm{TF}(w, m) \times \mathrm{IDF}(w, m). \tag{3}$$

We collected a set of words with highest TF-IDF scores in all messages of the data set. For each message, the TF-IDF scores of these words in the message were determined. These weights and additional features were assembled in a vector. In order to determine the TF-IDF scores we used Gensim, a Python and NumPy package for vector space modelling of text documents.

In addition we used the following features reflecting the syntactic structure of the messages:

- number of html tags in the message;
- number of links in the message;
- number of mismatched links, where the visible link is different from the hyperlink reference;
- number of scripts included in the message;
- number of tables in the message;
- number of embedded images;
- number of attachments to the message.

These features were assembled in an algebraic vector space model representing the data set. A number of independent initial clusterings were then obtained for the feature vectors of the messages in the sample using the following clustering algorithms.

3 Independent Initial Clusterings

The standard k-means clustering algorithm is described, for example, in [10], Section 3.3.2, and [24], Section 4.8. This algorithm randomly chooses k messages as centroids of clusters at the initialization stage. Every other message is allocated to the cluster of its nearest centroid. After that each iteration finds new centroids of all current clusters as a mean of all members of the cluster. This is equivalent to finding the point such that the sum of all distances from the new centroid to all other sequences in the cluster is minimal. Then the algorithm reallocates all points to the clusters of the new centroids. It proceeds iteratively until the centroids stabilize. The outcomes of the algorithm often depend on the initial selection of the very first centroids. We used the following two more advanced algorithms, which overcome this dependence.

First, we used the well-known *Global k-Means* algorithm, GKM, introduced in [17]. It overcomes the dependence on the initial choice of the centroids. It starts with just one centroid, which is taken as the mean of all points in the data set. Then the algorithm proceeds inductively. Suppose that i centroids have been found, for some $i < k$. Each of the given points in the data set is then chosen in turn and used as an $(i + 1)$-st initial centroid. For each of these choices of the additional centroids, the standard k-means algorithm is then run to partition the data set into $i + 1$ clusters. After that, all of the resulting partitions are compared with each other.

In order to evaluate each partition C, the algorithm uses the sum of squares of the distances from all points to the centroids of their clusters in the clustering C. This sum is denoted by $\mathrm{GKM}(C)$. The sum is taken over all clusters of the partition. Each cluster contributes the summand equal to the sum of all squared distances from all elements of the cluster to the centroid of the cluster. If a clustering $C = C_1 \dot\cup \ldots \dot\cup C_{i+1}$ is a disjoint union of the clusters $C_1, \ldots, C_{i+1}$, where each cluster C_j has a centroid m_j, then

$$\mathrm{GKM}(C) = \sum_{j=1}^{i+1} \sum_{x \in C_j} ||x - m_j||^2. \tag{4}$$

The partition which minimizes (4) is chosen as the best clustering with $i + 1$ clusters. The global k-means algorithm continues this process iteratively until it finds a partition into k clusters. Notice that the Modified Global k-means algorithm, MGKM, developed in [3], can be used to increase the efficiency of the GKM algorithm for large data sets.

Second, we used a modification of the *Multiple Start k-Means* algorithm, MSKM, considered in [3], [9] and [11]. The standard MSKM clustering algorithm selects many random sets of initial centroids, runs the k-means algorithm

for each of them, and chooses the partition minimizing the sum-of-squares objective function (4). The purpose of our investigation, however, is to find an optimal consensus among versatile clusterings. We have tried to include various different independent clusterings in the scheme. Since we have already included the global k-means algorithm minimizing the sum-of-squares (4), adding other algorithms concentrating on this objective function could skew the resulting outcome.

Instead, we modified the MSKM algorithm and used the following *Consensus Multiple Start k-Means* algorithm, CMSKM. It makes 50 random selections of the initial centroids, runs the standard k-means algorithm, and then finds an aggregated consensus clustering of the resulting 50 k-means clusterings. To find the consensus clustering we used the simplest cluster-based similarity partitioning algorithm, CSPA, described in [9]. It places two messages in the same cluster if they belong together to one cluster in the majority of the clusterings of the ensemble.

Third, we used a version of hierarchical agglomerative clustering algorithm known as the *Nearest Neighbour* clustering, NN, see [10], Section 3.3.7, [11] and [24], Section 4.7. It never merges large clusters, and only amalgamates separate messages, i.e. singleton clusters, to other clusters at each step. Given the number k of clusters to be found, it chooses k random messages as representatives of the clusters. For every other message m in the data set, it considers all messages which have already been assigned to the clusters and finds the nearest neighbour of m among these messages. The message m is then allocated to the cluster of its nearest neighbour. This continues until all messages are allocated to clusters. The outcome of the algorithm strongly depends on the initial random choice of the very first representatives. This is why it is very seldom used in this form.

In order to overcome the dependence on the initial random selection, as recommended in [9], we made 50 uniformly distributed selections of the initial representatives, run the nearest neighbour clustering algorithm for each of them, and then found a common consensus clustering of all the resulting clusterings, using the CSPA consensus aggregation algorithm again. Here we call the resulting procedure the *Consensus Multiple Start Nearest Neighbour* clustering algorithm, CMSNN.

Fourth, we looked at the *k-Committees* clustering method considered in [25] for a data set of DNA sequences. For a very small positive integer $r = 2, 3, ...$, it finds a very special set of r elements in each class, called the *committee of r representatives* of the class, or simply the *committee* of the class. The committee of a cluster C is defined as a set of r points $x_1, \ldots, x_r$ defined as a solution to the following optimization problem:

$$\text{minimize} \quad \max_{y \in C} \left(\min_{i=1,\ldots,r} ||x_i - y|| \right) \quad \text{subject to} \ \ x_1, \ldots, x_r \in C, \tag{5}$$

see [25], Section 5.

Every new message is then allocated to the class of its nearest committee member, see [25]. The training stage of the k-committees algorithm is not scalable, and this is why it has not been used in practice. However, after the completion

of the training, the algorithm runs fast. Our scheme makes it possible to apply this algorithm, since it has to be trained with a fairly small initial sample only.

In order to overcome the dependence of this algorithm on the initial choice of starting committees, we run it for 50 randomized selections of these representatives, and for small values of r from 2 to 6 representatives in each committee. Then we applied the CSPA consensus aggregation algorithm as above to combine the resulting ensemble into one clustering. In the present article this procedure is called the *Consensus Multiple Start k-Committees* algorithm, CMSKC.

4 Consensus Functions for Ensemble Clusterings

The process of finding the combined consensus clustering has also been divided into two substages. During the first substage several independent initial clusterings were ensembled using various consensus functions. This has produced a number of very similar consensus clusterings. During the second substage a fairly simple and fast consensus heuristic was used to combine them all into one common final consensus clustering.

During the first substage, given an ensemble of several independent clusterings on one and the same data set, consensus functions were applied to form new common consensus clusterings. Here we use the methods described, for example, in [5,20,23]. Let us denote the data set being investigated by

$$D = \{d_1, d_2, ..., d_n\}. \tag{6}$$

The clustering ensemble on this data set will be denoted by

$$C = \{C^{(1)}, C^{(2)}, \ldots, C^{(k)}\}, \tag{7}$$

where, for each clustering $C^{(i)}$, the whole set D is a disjoint union of the classes in this clustering so that

$$C^{(i)} = \{C_1^{(i)}, C_2^{(i)}, \ldots, C_{k_i}^{(i)}\}, \tag{8}$$

$$D = C_1^{(i)} \dot{\cup} C_2^{(i)} \dot{\cup} \ldots \dot{\cup} C_{k_i}^{(i)} \tag{9}$$

for all $i = 1, \ldots, k$.

Looking at the features described in Section 2, we applied several different clustering algorithms outlined in Section 3 to obtain initial clusterings. The following consensus functions and algorithms were invoked to combine the resulting cluster ensemble into one consensus clustering:

ALCH – Average Link Consensus Heuristic,
CBGF – Cluster-Based Graph Formulation,
CCPH – Consensus Clustering Pivot Heuristic,
HBGF – Hybrid Bipartite Graph Formulation,
IBGF – Instance-Based Graph Formulation,
KMCF – k-Means Consensus Function.

All these consensus clustering algorithms have been compared for numerous data sets in [5,6,20]. Here we include only a brief summary of these methods, and refer to [1], [5] and [9] for more details.

Average Link Consensus Heuristic, ALCH, is an agglomerative clustering algorithm described in [9]. It starts off with a partition where every element belongs to its own separate singleton cluster. For each pair of elements i, j, the proportion p_{ij} of the initial consensus clusterings which cluster i and j in different clusters is determined. Then the algorithm finds two clusters with the smallest average distance and merges them together into one new cluster. This is repeated until the two closest clusters have average distance greater than the set threshold $\tau = 1/4$.

Cluster-Based Graph Formulation, CBGF, is a graph-based consensus function. It defines a complete weighted undirected graph on the set of vertices consisting of all the given clusters. The weight of each edge of this graph is determined by a measure of similarity of the clusters corresponding to the vertices. Namely, for two clusters C' and C'' the weight of the edge (C', C'') can be set equal to

$$w((C', C'')) = \frac{|C' \cap C''|}{|C' \cup C''|}, \tag{10}$$

known as the *Jaccard index* or *Jaccard similarity coefficient*, see [21], Chapter 2. In order to ensure that clusters which have a lot of elements in common are grouped together, the edges with lowest weights are then eliminated by applying a graph partitioning algorithm. Each element is then allocated to the new final cluster where it occurs most frequently.

Consensus Clustering Pivot Heuristic, CCPH, is an agglomerative clustering algorithm described in [1]. It chooses a pivot element i uniformly at random from the unclustered elements. It finds all elements j such that the proportion p_{ij} of the given initial clusterings in the ensemble, which cluster i and j in different clusters, does not exceed the threshold value $\tau = 1/2$, and places all of these elements j in the same cluster with i. This continues until all elements are clustered.

Hybrid Bipartite Graph Formulation, HBGF, is a consensus function based on a bipartite graph. It has two sets of vertices: clusters and elements of the data set. A cluster C and an element d are connected by an edge in this bipartite graph if and only if d belongs to C. (The weights associated to these edges may have to be chosen as very large constants if the particular graph partitioning algorithm does not allow zero weights and can handle only complete graphs.) An appropriate graph partitioning algorithm is then applied to the whole bipartite graph, and the final clustering is determined by the way it partitions all elements of the data set. We used METIS graph partitioning software described in [14].

Instance-Based Graph Formulation, IBGF, is also a consensus function based on a complete undirected weighted graph. Vertices of the graph are all elements of the data set. The edge (d', d'') has weight given by the formula

$$w((d', d'')) = \sum_{i=1,\ldots,k;\ C_i(d')=C_i(d'')} 1/k,$$

where $C_i(x)$ stands for the cluster containing x in the i-th clustering. This means that $w((d', d''))$ is the proportion of clusterings where the clusters of d' and d'' coincide. Then IBGF applies an appropriate graph partitioning algorithm to divide the graph into classes. These classes determine clusters of the final consensus clustering.

k-Means Consensus Function, KMCF, relies on the standard k-means algorithm to produce final clustering. A complete explanation of this method is given in [23]. KMCF uses the set of all clusters in all clusterings of the ensemble as features for its feature vectors. For each element $d \in D$ and each cluster $C_j^{(i)}$, the $C_j^{(i)}$-th component of the feature vector of d is set to 1 if d belongs to $C_j^{(i)}$, and it is set to 0 otherwise. The standard k-means clustering algorithm is then used to cluster this set of feature vectors in order to find the consensus clustering.

During the second substage all the resulting consensus clusterings described above have been combined into one common consensus clustering using a very simple Majority Rule heuristic described in [9]. It is also known as the quote rule, see [8]. The Majority Rule is an agglomerative clustering algorithm, which starts with a partition where every element belongs to a separate singleton cluster. For each pair of elements i and j it computes the proportion p_{ij} of the initial consensus clusterings which cluster i and j in different clusters. If p_{ij} is less than a threshold value τ, then the current clusters containing i and j are combined together into one cluster. In our problem we used τ equal to the half of the total number of the consensus clusterings being combined.

5 Supervised Classification Algorithms

The resulting consensus clustering described in Section 4 was used to train the following fast supervised classification algorithms.

First, we used the k-means classification algorithm considered, for example, in [25] for a data set of DNA sequences. It finds the centroids of all clusters in the training set, and then allocates every new message to the cluster of its nearest centroid (see [4], Chapter 4 and Section 10.4.3, [24], Chapter 4, and [22]). We have incorporated this method into our scheme, because it is very fast.

Second, we used the simplest and fastest nearest neighbour classification algorithm. It utilizes the clusters of the training set, called the *prototypes* or *exemplars*, see [4], Chaper 4, and [24], Section 6.4. The algorithm allocates every new message to the class of its nearest exemplar.

Finally, we used the k-committees classification algorithm considered in [25] for a data set of DNA sequences, see also [13]. For a very small positive integer r, it also finds a *committee* of the class. The committee of C is again defined as a set of points $x_1, \ldots, x_r$, which are found as a solution to the optimization problem (5). The committees have to be found only for the small training set created

by the consensus clustering. Training stage of the k-committees algorithm is not scalable. In our scheme the algorithm only has to be trained on the relatively small training set, and this is why it can be incorporated in the scheme. In order to classify new messages after the training, the algorithm allocates every new message to the class of its nearest committee member, and can be executed very fast, since the number of the representatives in each committee is a very small and fixed positive integer.

6 Experimental Results

We have carried out experimental investigation of the novel approach to clustering for a sample of 3276 emails randomly selected from a very large data set of emails supplied by the industry partners of the Centre for Informatics and Applied Optimization. All of these emails have already been classified as phishing messages by the information security group of our industry partners. Many of these emails contain both text and hyperlinks and include HTML script, tables and images.

A flexible preprocessing and feature extraction system has been implemented in Python for the purposes of this investigation. It has been used to extract features concerning the content and structure of the emails, and hyperlinks embedded in the text as described in Section 2.

First, we found combined consensus clustering for the whole data set, following the procedure described in Section 4. Second, this clustering was used as a benchmark to determine the accuracy of the performance of several classification algorithms incorporated into the scheme. We used ten times tenfold cross validation to evaluate the accuracy of our multistage scheme. Each of the ten times, the data set was divides into ten equal parts, nine parts were used as a training set, and one part was used as a testing set. We run the combined consensus clustering procedure on the training set to prepare input for training as described in Section 4. After that the supervised classification algorithms described in Section 5 were trained on the training set obtained. Our experimental results compare the efficiency of the performance of these classification

Table 1. Ten times tenfold cross validation

Accuracy of classification algorithms w.r.t. consensus clustering				
Algorithm	Number of clusters			
	5	10	15	20
k-means	66.50	60.07	57.69	52.28
k-committees with $r = 2$	76.11	72.56	68.12	61.12
$r = 3$	83.23	77.44	73.24	70.58
$r = 4$	88.05	83.66	80.05	74.84
$r = 5$	91.35	87.86	80.07	78.49
$r = 6$	93.55	86.21	84.98	78.44
Nearest Neighbour	88.75	81.47	77.25	73.45

algorithms presented in Section 5 after their training on the initial consensus clustering data.

We used ten times tenfold cross validation to evaluate the accuracy of these algorithms. The accuracy of their performance was then evaluated in comparison with the combined overall total consensus clustering obtained previously. It is defined as the percentage of the messages in the test set which are classified correctly. This was repeated ten times, and the average accuracy was then calculated for each algorithm. The results of our experiments are summarized in Table 1.

7 Conclusion

This article looked at a novel method for profiling phishing emails. First, a multitude of independent clustering algorithms were used to a randomized sample of the messages. Second, several consensus functions and sophisticated algorithms were applied to combine these independent clusterings into one final consensus clustering. Third, several fast supervised classification algorithms were trained on the consensus clustering of the randomized sample. Finally, these fast classification algorithms classified the whole data set. This approach facilitates the inclusion of contributions from domain experts via adjusting the training set created by consensus clustering. We applied this approach to a set of phishing emails provided by the industry partners of the Centre for Informatics and Applied Optimization. The experimental results show that the nearest neighbour and the k-committees algorithms achieve much better accuracy in this scheme, compared with the k-means algorithm, and the k-committees algorithm outperforms the nearest neighbour on the average. It has also been demonstrated that the scheme can be used in practice. If required, then it has the potential to facilitate the inclusion of contributions form domain experts for adjusting the training set produced by consensus clustering algorithms. Another advantage of our approach is in its ability to combine highly accurate consensus clustering techniques with fast and simple classification algorithms in one scheme.

Acknowledgements

The authors are grateful to three referees for comments and corrections that have helped to improve the text of this article.

References

1. Ailon, N., Charikar, M., Newman, A.: Aggregating inconsistent information: ranking and clustering. In: Proc. 37th Annual ACM Symposium on Theory of Computing, pp. 684–693 (2005)
2. Anti-Phishing Working Group (2009), http://apwg.org/ (retrieved April 2010)

3. Bagirov, A.M.: Modified global k-means algorithm for minimum sum-of-squares clustering problems. Pattern Recognition 41, 3192–3199 (2008)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, Hoboken (2001)
5. Fern, X.Z., Brodley, C.E.: Cluster ensembles for high dimensional clustering: an empirical study. J. Machine Learning Research (2004)
6. Fern, X.Z., Brodley, C.E.: Solving cluster ensemble problems by bipartite graph partitioning. In: Proc. 21st Internat. Conference on Machine Learning, ICML 2004, Banff, Alberta, Canada, July 4-8, vol. 69, p. 36. ACM, New York (2004)
7. Fette, I., Sadeh, N., Tomasic, A.: Learning to detect phishing emails. In: Proc. 16th Internat. Conf. on the World Wide Web, WWW 2007, pp. 649–656. ACM Press, New York (2007)
8. Filkov, V., Skiena, S.: Heterogeneous data integration with the consensus clustering formalism. In: Proc. of Data Integration in the Life Sciences, pp. 110–123 (2004)
9. Goder, A., Filkov, V.: Consensus clustering algorithms: comparison and refinement. In: Proc. Tenth SIAM Workshop on Algorithm Engineering and Experiments, ALENEX 2008, pp. 109–117 (2008)
10. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs (1988)
11. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31(3), 264–323 (1999)
12. Joachims, T.: A probabilistic analysis of the Rocchio algorithm with TF-IDF for text categorization. In: Proc. 14th Internat. Conf. Machine Learning, pp. 143–151 (1997)
13. Kang, B.H., Kelarev, A.V., Sale, A.H.J., Williams, R.N.: A new model for classifying DNA code inspired by neural networks and FSA. In: Hoffmann, A., Kang, B.-h., Richards, D., Tsumoto, S. (eds.) PKAW 2006. LNCS (LNAI), vol. 4303, pp. 187–198. Springer, Heidelberg (2006)
14. Karypis, G., Kumar, V.: METIS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices, Technical Report, University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Centre, Minneapolis (1998)
15. Layton, R., Watters, P.: Determining provenance in phishing websites using automated conceptual analysis. In: Proc. 4th Annual APWG eCrime Researchers Summit, Tacoma, WA (2009)
16. Layton, R., Brown, S., Watters, P.: Using differencing to increase distinctiveness for phishing website clustering. In: Proc. Cybercrime and Trustworthy Computing Workshop, CTC 2009, Brisbane, Australia (2009)
17. Likas, A., Vlassis, N., Verbeek, J.J.: The global k-means clustering algorithm. Pattern Recognition 36, 451–461 (2003)
18. Ma, L., Yearwood, J., Watters, P.A.: Establishing phishing provenance using orthographic features, APWG E-crime Research Summit (2009)
19. OECD Task Force on Spam, OECD Anti-Spam Toolkit and its Annexes (2009), http://www.oecd-antispam.org/ (retrieved April 2010)
20. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. J. Machine Learning Research 3, 583–617 (2002)
21. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison Wesley, Reading (2005)

22. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 4th edn. Academic Press, London (2008)
23. Topchy, A., Jain, A.K., Punch, W.: Combining multiple weak clusterings. In: Proc. IEEE Internat. Conf. on Data Mining, pp. 331–338 (2003)
24. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Elsevier/Morgan Kaufman, Amsterdam (2005)
25. Yearwood, J.L., Kang, B.H., Kelarev, A.V.: Experimental investigation of classification algorithms for ITS dataset. In: PKAW 2008, Pacific Rim Knowledge Acquisition Workshop, Hanoi, Vietnam, December 15-16, pp. 262–272 (2008)

People Recommendation Based on Aggregated Bidirectional Intentions in Social Network Site

Yang Sok Kim, Ashesh Mahidadia, Paul Compton, Xiongcai Cai, Mike Bain, Alfred Krzywicki, and Wayne Wobcke

School of Computer Science and Engineering,
The University of New South Wales, Sydney, NSW, 2052, Australia
{yskim,ashesh,compton,xcai,mike,alfredk,wobcke}@cse.unsw.edu.au

Abstract. In a typical social network site, a sender initiates an interaction by sending a message to a recipient, and the recipient can decide whether or not to send a positive or negative reply. Typically a sender tries to find recipients based on his/her likings, and hopes that they will respond positively. We examined historical data from a large commercial social network site, and discovered that a baseline success rate using such a traditional approach was only 16.7%. In this paper, we propose and evaluate a new recommendation method that considers a sender's interest, along with the interest of recipients in the sender while generating recommendations. The method uses user profiles of senders and recipients, along with past data on historical interactions. The method uses a weighted harmonic mean-based aggregation function to integrate "interest of senders" and "interest of recipients in the sender". We evaluated the method on datasets from the social network site, and the results are very promising (improvement of up to 36% in success rate).

Keywords: Recommender systems, Social networks.

1 Introduction

A traditional recommender system considers a user's preferences along with characteristics of objects being recommended to suggest objects that a user might be interested in. Here, objects could be books, music albums, consumer goods, etc. It is not always easy to develop a good recommender system because such a system should take into account the hidden preferences of a user and often the special characteristics of objects. Such systems are based on one-way interaction model that assumes that objects that are being recommended are not active, and therefore the "intentions" of an object are not considered while making recommendations. However, when recommending people to a given user in a social network site, we also need to consider whether a person recommended is likely to "like" a given user or not. In other words, here objects being recommended are not "passive", and importantly we need to consider the "intentions" of a person being recommended to a given user, along with the "intentions" of a given user. We call this kind of interaction a two-way interaction model. A recommender system for a two-way interaction model needs to consider bidirectional intentions while generating recommendations.

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 247–260, 2010.

In this research, we propose a new recommendation method for a two-way interaction model. We classify users into two types – senders and recipients – according to their behaviours. For each sender, the method creates a recommendation rule using profiles of senders and recipients, along with interaction history of all users. The method combines the "*interests of senders*" and the "*interest of recipients in the senders*" using a criteria *aggregation function*. The importance of these two interests is represented by weights. These weights could be heuristically or experimentally decided to create effective recommendations. Experimental results show that the acquired rules are significantly different for different weights for "*interests of senders*" and "*interest of recipients in the senders*". We classified users into four groups based on the number of interactions they have received, and analyse their performance.

This paper is organized as follows: Section 2 summarizes the literature review on recommender systems and multi-criteria aggregation research. Section 3 explains our rule learning method and Section 4 explains the evaluation method. Experimental results are summarized in Section 5. Conclusions and further work are discussed in Section 6.

2 Literature Review

2.1 Recommender Systems

Recommender systems usually propose candidate items/objects to a given user by using similar users' preference patterns. For example, if user 1 and user 2 are similar and user 1 purchases item A, the system suggests item A to user 2. Here, defining similarity of users is a critical problem that a recommender system must address. Often user profiles and behaviour are used to define similarity. **Profile-based methods** use user profiles to calculate similarity between users. Various machine learning techniques, such as decision trees, rule induction, nearest neighbour, and Naïve Bayes classification, are usually employed for this purpose [1]. Profile-based recommenders may not work well when user profiles are not sufficient for learning user similarity[2]. **Behaviour-based methods** use users' behaviour to calculate similarity. For example, the Amazon recommender system [3] uses view or purchase history to identify similar users. Many other item-to-item collaborative filtering recommender systems [4-6] and social network-based recommenders [7-10], use users' behaviour to identify user similarity. We employ behaviour-based methods as well as profile-based methods. The social network site we examined provides us with a large dataset outlining user profiles for a very large number of users, and past interactions between users.

2.2 Bidirectional Criteria in Recommender Systems

As discussed earlier, we need to consider bidirectional intentions for our recommendations. It is important to consider how to combine two different "*intentions*", before making recommendations. Though multi-criteria decision making has been extensively researched [11], it has not received attention in the recommendation research community. Several researchers discussed multi-criteria in

relation to rating problems of collaborative filtering based recommenders [12-14]. Even though our research is not based on the collaborative filtering method, it is necessary to reflect two different intentions, namely the intentions of senders and the recipients and thus it is necessary to consider their aggregation. Aggregation functions could be obtained by domain expertise, statistical techniques, and/or machine learning techniques [12]. We used a weighted harmonic mean based aggregation function, because "*interest of senders*" and "*interest of recipients in the sender*" are ratios and typically a weighted harmonic mean is appropriate for situations where a weighted average of rates is desired.

3 Recommendation Rule Acquisition Method

3.1 Definitions

A **user** is represented by M attribute values. A **sender** is a user who initiates an interaction and a **recipient** is a user who receives an interaction from a sender.

A **subgroup** is a group of users where at least m attribute have the same values, here $1 \leq m \leq M$. A **basic subgroup** is a subgroup where m equals 1. For example, the set of users belonging to the same ethnic group is a subgroup.

An **interaction** is an action where a sender sends a message to a recipient. A **sending interaction** is an interaction where a sender has not received an interaction from the recipient. In other words, a sender is initiating a new conversation. A sending interaction is represented by an arrow (→). A **responding interaction** is an interaction where a sender has already received an interaction from the recipient. In other words, this represents a reply from a recipient of an interaction. A responding interaction is represented by an arrow (←). The number of sending interactions from a subgroup of senders (S_i) to all the recipients (R) is denoted as $\mathrm{ns}(S_i \rightarrow R)$, and the number of sending interactions from a subgroup of senders (S_i) to a subgroup of recipients (R_i) is denoted as $\mathrm{ns}(S_i \rightarrow R_j)$. Whereas a sending message usually shows only a positive intention of a sender, a responding message exhibits both positive and negative intentions of a recipient. The number of positive responding interactions from a subgroup of recipients (R_i) to a sender subgroup (S_i) is denoted as $\mathrm{ns}(\mathrm{S_i} \leftarrow \mathrm{R_j}(+))$. Similarly, the number of negative responding interactions from a subgroup of recipients (R_i) to a sender subgroup (S_i) is denoted as $\mathrm{ns}(\mathrm{S_i} \leftarrow \mathrm{R_j}(-))$.

Interest of Senders: For a sender subgroup ($\mathrm{S_i}$), its interest in a recipient subgroup ($\mathrm{R_i}$) is defined as follows:

$$\mathrm{I}(\mathrm{S_i}, \mathrm{R_j}) = \mathrm{ns}(\mathrm{S_i} \rightarrow \mathrm{R_j}) / \mathrm{ns}(\mathrm{S_i} \rightarrow \mathrm{R}), \quad (1)$$

where $\mathrm{ns}(\mathrm{S_i} \rightarrow \mathrm{R_j})$ represents the number of interactions sent from the sender subgroup ($\mathrm{S_i}$) to the recipient subgroup ($\mathrm{R_j}$) and $\mathrm{ns}(\mathrm{S_i} \rightarrow \mathrm{R})$ represents the number of interactions sent from the sender subgroup ($\mathrm{S_i}$) to all recipients (R). This measure represents how much the sender subgroup ($\mathrm{S_i}$) is interested in the recipient subgroup ($\mathrm{R_j}$), compared to the rest of the recipients in R.

Interest of Recipients in Senders: For a recipient subgroup (R_i), its interest in a sender subgroup (S_i) is defined as follows:

$$I(R_j, S_i) = ns(S_i \leftarrow R_j(+))/ns(S_i \rightarrow R_j), \quad (2)$$

where $ns(S_i \rightarrow R_j)$ represents the number of interactions sent from the sender subgroup (S_i) to the recipient subgroup (R_j) and $ns(S_i \leftarrow R_j(+))$ represents the number of positive responses sent from the recipient subgroup (R_j) to the sender subgroup (S_i). This measure represents how much the recipient subgroup (R_j) is interested in the sender subgroup (S_i).

3.2 Interaction Look-Up Table

As we modelled our recommendation method using profiles and behaviours of users, each user's profile and the log of interactions between users were collected from a specified training period. Based on these data, an interaction look-up table for each attribute was created for rule learning. Fig. 1 illustrates such an interaction look-up table for a single attribute.

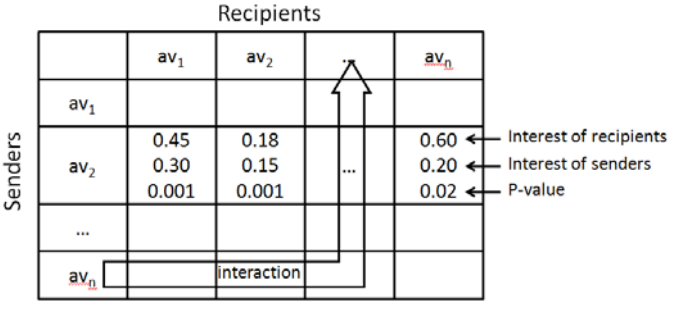


Fig. 1. Interaction Look-up Table between Attribute Values

Each row represents senders with a particular attribute value, and each column represents recipients with a particular attribute value. For example, for an interaction look-up table for the attribute "Ethnicity", different rows would represent senders of different ethnicities (Greek, Vietnamese, English, etc.), and similarly different columns would represent recipients of different ethnicities. In this interaction look-up table, each cell contains *interaction measures* between the senders in the corresponding row and the recipients in the corresponding column. There is a major simplifying assumption in this, that attributes are independent and that the interaction between senders and recipients can be considered separately for each attribute.

3.3 Best Matching Pair Discovery

For each attribute of a given sender, the method attempts to find a best matching value of the same attribute for recipients, such that the sender is likely to be interested in the recipients, and the recipients are more likely to reply positively. For example,

"Australian" senders send 80% of their interactions to "English" recipients, and 20% of their interactions to "German" recipients. This clearly indicates that "Australian" senders are more interested in "English" recipients than "German" recipients. Now, suppose 40% of the "English" recipients reply positively, and 70% of the "German" recipients reply positively. If we want to recommend recipients based on the "interest of senders" (see Section 3.1), we should recommend "English" recipients. However, if we want to recommend recipients based on the "interest of recipients in senders", we should recommend "German" recipients. This example illustrates that each criterion only partially captures the interest of the sender and recipients. We need to combine these two criteria in order to generate recommendations such that both a sender and the corresponding recipients are interested in each other.

In this research, we used the following *interest aggregation* function to integrate a sender's interest in recipients ("*interest of senders*", see Section 3.1) and recipients' interest in senders ("*interest of recipients in the sender*", see Section 3.1).

Interest Aggregation Function: A weighted harmonic mean, $H(S_i, R_j)$, was employed as an *interest aggregation* function, which is defined as follows:

$$\mathrm{H}(\mathrm{S_i}, \mathrm{R_j}) = \frac{\omega_s + \omega_r}{\frac{\omega_s}{I(S_i,R_j)} + \frac{\omega_r}{I(R_j,S_i)}} \tag{3}$$

where $I(S_i, R_j)$ and $I(R_j, S_i)$ represent the interest of senders and the interest of recipients in senders respectively (see Section 3.1), ω_s represents a weight for "*interest of senders*" and ω_r represents a weight for "*interest of recipients in the sender*". The sum of these two weights ($\omega_s + \omega_r$) is 1.

For a given sender and a value of ω_s, the method calculates weighted harmonic mean for each value of the same attribute for recipients using Eq (3). The method selects an attribute value for recipients with the highest weighted harmonic mean as the best matching pair for that attribute. In our experiments, various combinations of the weights were evaluated.

3.4 Recommendation Rule Acquisition

For a given sender, the method finds best matching pairs for every attribute. From these best matching pairs, all the recipient attribute values can be collected to formulate a rule that could be used to generate possible recommendations. For example, the following rule (only few conditions are displayed here) could be used to generate recommendations for a given sender,

```
Gender = Female
AND Job = Accounting
AND Location = Adelaide
AND Age = 40 ~ 44
. . . .
. . . .
```

However, in practice such a rule with all the recipient attribute values may prove too specific and may not generate any recommendations. If such a rule could not generate the required number of recommendations, the method relaxes the rule by removing an attribute from a rule that has the lowest value for "*interest of recipients in senders*".

The process is repeated and the rule is relaxed until we can generate sufficient recommendations to satisfy the constraints discussed in Section 3.5.

In the process of selecting an attribute for relaxing a current rule, we did not use criteria such as "*interest of senders*" (see Section 3.1) and "*interest aggregation value*" (see Section 3.2) because they are influenced by the number of possible values of an attribute, and therefore we could not compare them across different attributes.

3.5 Constraints for Candidate Generation

We are only interested in candidates who are active users. We define an active user in three ways:

- The user joined the social networking website recently,
- Or the user sent initiating interactions recently,
- Or the user received and viewed initiating interactions from others recently. (The log identifies if a recipient looks at a contact message).

We define 'recently' as the last month. Preliminary data analysis on temporal activity showed that user activity in the past month provides a good indication of how likely they are to respond.

We are also interested in discovering subgroups of senders and recipients such that the interaction behaviour between them is statistically significant. We could use the current rule to generate a subgroup of recipients, and the corresponding subgroup of senders (based on the corresponding best matching pairs). We calculate the probabilistic significance of the interaction behaviour between these two subgroups using the following binomial formula:

$$\mathbf{P(r)_{binomial} = nCr \times p^{r} \times q^{n-r}} \tag{4}$$

where n is the number of sending interactions between the two groups, r is the number of positive replies between the two groups, p is the base success rate between all senders and recipients (no of all positive replies/no of all sending interactions), q is 1-p. We consider an interaction behaviour is significant if the p-value is < 0.05.

4 Experimental Design

4.1 Data Sets

We used data sets obtained from a large commercial social network site. User profile data contained 32 attributes for each user, such as age, location, ethnic background, physical appearance (body type, hair colour, etc.), occupation industry and level, children and marital status, and others. All numeric attribute values such as age, number of photos, number of children, etc were transformed to nominal values. User interaction log data contains the interaction history between senders and its corresponding recipients. Each log entry identifies a sender, the corresponding recipient and the reply message. Reply messages are classified into positive and negative, so that each interaction is also classified as a positive or negative interaction. A failure to reply was taken as a negative interaction. This interaction log

data is used to calculate interest measurers between a sender subgroup and its corresponding recipient subgroup (see Section 3.1). Two sets of training data, **Train I** and **Train II**, and one set of test data, **Test**, were collected for the experiment. The data sets used for this research are summarised in Table 1. **Train I** was used for our rule learning method. It contains interaction history data for three months and was used to generate the interaction look-up tables and also to calculate "interest of senders" (see Section 3.1), "interest of recipients in the senders" (see Section 3.1), and p-value for significance test (see Section 3.5) in the rule learning process. **Train II** was collected for the Collaborative Filtering (CF)-based method from March, 2009 (one month). Our preliminary data analysis using the CF method over different time periods showed that a training period of one month was appropriate. **Test** data were collected from the first week of April for evaluation. This was immediately following the CF training period to give the CF method the best chance of performing.

Table 1. Training and Test Data Set

Data Set	Total interaction	Positive Interaction	Negative Interaction	Success Rate
Train I	*3,888,034*	*689,419*	*3,198,615*	*17.7%*
Train II	*1,357,432*	*236,521*	*1,120,911*	*17.4%*
Test	*284,702*	*47,468*	*237,234*	*16.7%*

We examined whether or not the number of interactions received influences recommendation performance and recommendation rule acquisition. All senders in the test period (30,387) were classified into four types of senders based on the number of interactions received during March, 2009. Note that zero-received sender in the training period may receive interactions in the test period.

Table 2. Sender Types Based on Interaction Received

Sender Type	Interaction Received (n)	Users (%)
zero-received	$n = 0$	7,560 (25%)
few-received	$1 \leq n \leq 3$	8,507(28%)
Average	$4 \leq n \leq 20$	11,223(37%)
Popular	$20 < n$	3,097(10%)

Recommendations were generated using different weights for "*interest of senders*" (ω_s) and "*interest of recipients in the sender*" (ω_r). We used 0, 0.25, 0.50, and 0.75 for ω_r and the corresponding ω_s is 1.0, 0.75, 0.5, and 0.25, as $\omega_s + \omega_r$ is 1 (see Section 3.3). In the following discussion, unless otherwise indicated, "*weight*" means "*interest of recipients in the sender*" (ω_r). Weight 0 was employed to evaluate an extreme situation, where only "*interest of senders*" is considered for rule learning. Weight 0.5 was employed to evaluate when the two interests are regarded to have same importance. Weight 0.25 and weight 0.75 (ω_r equal 0.25) were selected to see the results if one of the two interests has more importance than another. Weight 1.00,

which is another extreme case that considers only "*interest of recipients in the sender*", was not employed, because senders in the testing period rarely sent interactions to the recipients recommended by our method when weight is 1.00. This is understandable because weight 1.00 does not consider "*interest of senders*".

4.2 Collaborative Filtering

Interactions are likely to depend significantly on an individual's appearance in photos and other personal preferences which may be included in free text, but are not captured in the attribute data we used. In this case, Collaborative Filtering (CF) could be used to generate recommendations. We implemented a CF method based on [1] and compared it with our approaches. In a typical CF model, two items are similar if they have been purchased together by a large number of customers, and the unpurchased item is suggested to a user if he/she purchases the other similar item. Based on these criteria, two users are considered to be similar senders to the extent that they have sent contacts to the same recipients. For example, if sender s_1 sends message to recipient r_1, r_2, and r_3 and sender s_2 is to recipient r_1 and r_2, they are regarded as similar senders, because they both sent interactions to recipient r_1 and r_2. If a new user u sent contacts to r_1, then u is also similar to s_1 and s_2, so r_1, r_2 and r_3 can be recommended to u. Note that the rank of r_1, r_2 and r_3 may be differ because r_1 and r_2 are recommended by both s_1 and s_2, whereas s_3 is only recommended by s_1. For the evaluation, the test set was checked to see whether the interactions between a user and the candidates suggested by our recommendation methods and the CF method had actually occurred. As this was a retrospective study, we could not assess what would happen if users had followed our recommendations. All we could check was whether they had a higher success rate if they had happened to contact a person we would have recommended. Very popular users, who received more than 50 contacts in March, 2009, were not considered for evaluation.

4.3 Evaluation Metrics

We used the following metrics and variables to assess our method:

- Θ: a given recommendation method,
- M: the senders who were active in March, 2009 and who sent interactions in the test period,
- N: the senders who were members of M and who would have received suggestions by Θ,
- O: all interactions suggested by Θ (representing a sender and a predicted recipient) for all senders in N, and
- Q: all interactions in the testing period by all senders in M, and
- K: the intersection between O and Q

For the performance evaluation of each method, the following metrics were used:

Coverage: The proportion of N from M in the test period, i.e.

$$Cov = \mathrm{n}(N)/\mathrm{n}(M) \tag{5}$$

where n(M) is the number of M and n(N) is the number of N.

Success Rate: The proportion of those predicted successful contacts of *K*, i.e.

$$Succ = ns(K,+)/ns(K) \tag{6}$$

where $ns(K)$ is the number of interactions of K and $ns(K,+)$ is the number of positive interactions of *K*.

For the rule acquisition evaluation, the following metrics were used:

Rule Usage: The number of users per rule, i.e.

$$Usage = \mathrm{n}(N)/\mathrm{n}(R) \tag{7}$$

where $\mathrm{n}(N)$ is the number of users covered by a given method and $\mathrm{n}(R)$ is the number of rules from that method.

Condition Complexity: The number of condition elements per rule, i.e.

$$Complexity = \mathrm{n}(C)/\mathrm{n}(R) \tag{8}$$

where $\mathrm{n}(C)$ is the number of condition elements of all rules from given method and $\mathrm{n}(R)$ is the number of rules for that method.

5 Results

5.1 Coverage

On average the CF method can suggest recommendations for 74.0% of all senders in the test period (see Table 3). Whereas the CF method could suggest many recommendations for the senders who received many interactions in the recent month (March, 2009), it could suggest fewer recommendations for the senders who did not receive many interactions in the recent month. It suggests recommendations for 77.3% of the *few-received* senders, 86.7% of *average* senders, and 89.2% of *popular* senders in the testing period, but it only suggests recommendations for 45.4% of *zero-received* senders. However, our method can generate recommendations for all the senders in the testing period.

Table 3. Recommended Candidates Ratios

Sender Type	CF (Coverage)	Rule (Coverage)
Zero-received	*3,431 (45.4%)*	7,560 (100.0%)
Few-received	6,572 (77.3%)	8,507 (100.0%)
Average	9,725 (86.7%)	11,223 (100.0%)
Popular	2,767 (89.2%)	3,097 (100.0%)
All users	22,495 (74.0%)	30,387 (100.0%)

5.2 Success Rate

Success rate results are summarized in Table 4. For the test period interactions, the overall baseline success rate is 16.7%. *Popular* senders (24.8%) show the highest baseline success rate, followed by *average* senders (18.2%), *zero-received* senders

(14.1%), and *few-received* senders (14.0%). In some way, as the number of received interactions represents the popularity of the senders, this result is consistent with common sense that popular users receive more positive replies than less popular users.

Table 4. Success Rate

Sender Types	Test	CF	Rule with different weight for ω_r			
			0.00	0.25	0.50	0.75
Zero-received	*14.1%*	*11.0%*	*18.1%*	*18.5%*	*18.9%*	*20.0%*
Few-received	*14.0%*	*13.8%*	*20.2%*	*21.3%*	*20.9%*	*20.6%*
Average	*18.2%*	*19.3%*	*23.6%*	*24.0%*	*24.8%*	*25.3%*
Popular	*24.8%*	*25.8%*	*30.8%*	*30.4%*	*30.5%*	*30.3%*
All users	*16.7%*	*17.3%*	*21.5%*	*22.0%*	*22.2%*	*22.6%*

The success rate of the CF method (17.3%) is slightly higher than that of the testing baseline success rate. Similarly it is slightly higher for popular and average senders, and more significantly lower for zero-received. Our method gives higher success rates for all types of users (21.5% ~ 22.6%) (see Table 4), compare to both the baseline rates and the CF method. This is caused by the fact that whereas the CF method only reflects "interests of senders", our method reflects both "interests of senders" and "interests of recipient in the sender". The success rates increase as the weights increase from 0.00 to 0.75, but it is not significant. Success rate improvement for each sender types is summarized in Table 5, where success rate improvement values obtained by dividing each success rate by its corresponding baseline success rate of the testing period. The results show that our method is more effective for the "*few-received*" senders, followed by "*zero-received*", "*average*", and "*popular*" senders. This result is interesting because less popular users generally need more help from recommender systems than popular users.

Table 5. Success Rate Improvement

Sender Types	Test	CF	Rule with different weight for ω_r			
			0.00	0.25	0.50	0.75
Zero-received	*1.00*	*0.78*	*1.28*	*1.31*	*1.34*	*1.42*
Few-received	*1.00*	*0.98*	*1.44*	*1.52*	*1.49*	*1.47*
Average	*1.00*	*1.06*	*1.30*	*1.32*	*1.36*	*1.39*
Popular	*1.00*	*1.04*	*1.24*	*1.22*	*1.23*	*1.22*
All users	*1.00*	*1.04*	*1.29*	*1.32*	*1.33*	*1.36*

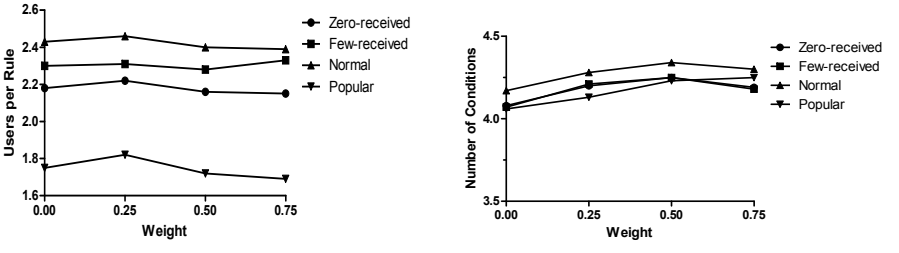
5.3 Rule Acquisition

Our method generates a rule for each user to produce the most appropriate recommendations, so the number of rules should be the same as the number of senders in the testing period. However, as the same rules can be generated for different senders, far fewer rules are created as summarized in Table 6. The average

number of users per rule is illustrated in Fig. 2 (a), showing less "*popular*" senders and "*zero-received*" senders per rule than "*few-received*" and "*normal*" senders. This means more rules are created for "*popular*" senders and "*zero-received*" senders compared to "*few-received*" and "*normal*" senders. This may be caused by the fact that the rules for "*popular*" senders and "*zero-received*" senders may be more complex than others. However, the data does not show that the rules for "*popular*" senders and "*zero-received*" senders use more conditions than rules for other types of users (see Fig. 2 (b)).

Table 6. Rule Acquisition Results

Sender Types	Rule with different weight for ω_r			
	0.00	0.25	0.50	0.75
Zero-received	3,470	3,408	3,503	3,509
Few-received	3,705	3,684	3,733	3,656
Average	4,617	4,557	4,686	4,697
Popular	1,767	1,702	1,800	1,830
All users	8,910	8,538	8,766	8,685



(a)Average Users per Rule (b)Conditions per Rule

Fig. 2. Rule Acquisition

Table 7 shows some example users where different weights produce different rules. For example, for the user id 1074, three different rules (rule ids 24, 14574, 23897) are created to recommend possible recipients. Taking user 1074 as an example, the first column in Table 8 lists some of the attribute values of the sender, and the second column lists three different rules (based on different weights) for generating possible recommendations. When the weight is zero, female recipients who live in "**`Adelaide`**" and whose job is in "**`Healthcare / Medical`**" are proposed as recommendations. As shown in Table 9, the senders who live in "**`Adelaide`**" sent 84% of their interactions to recipients in "**`Adelaide`**", and 30% of these interactions result in positive replies. Similarly, senders who "**`have children living at home sometimes`**" sent 34% of their interactions to recipients who "**`have children living at home`**" and 32% of these interactions result in positive replies. When the weight is zero, the method only considers sender's interest and ignores recipients' interest in senders. Therefore, the selection of attributes is biased towards higher "*interest of senders*" values. For example, **`location=Adelaide`** is selected due to its higher "Interest of Senders" value, when ω_r=0. However, when the

weight is 0.75, more emphasis is given to the values of "*interest of recipients in the sender*" while selecting attributes. For example, "**location=Adelaide**" is not selected due to its lower "*interest of recipients in the sender*" value compare to other three attributes listed in the rule for ω_r=0.75.

Table 7. Examples of the Recommended Rule Changes

User ID	Applied Rule ID with different weight for ω_r			
	0.00	0.25	0.50	0.75
98	3	3	3	23892
462	8	14562	14562	23894
735	13	14	14	14
1074	**24**	**14574**	**14574**	**23897**
1166	26	14578	19670	23899
1364	28	14581	19673	23900

Table 8. Example Rules of User 1074

Senders		Rules for recommended recipients
	Rule ID	Condition
Gender = Male Age = 40~44 Job = Property/Real Estate Location = Adelaide Have Children = Yes, have children living at home sometimes	24 (ω_r=0)	Gender = Female AND Location = Adelaide AND Job = Healthcare / Medical
	14574 (ω_r=0.25 or 0.5)	Gender = Female **AND Have Children=Yes, have children living at home** AND Job = Healthcare / Medical AND Location = Adelaide
	23897 (ω_r=0.75)	Gender = Female AND Have Children=Yes, have children living at home AND Job = Healthcare / Medical **AND Age = 40 ~ 44**

Table 9. Interest of Senders and Interests of Recipients in the Sender

Attribute	Attribute Value		Interest of Senders	Interest of Recipients in the Sender
	Male	Female		
Location	Adelaide	Adelaide	84%	30%
Have Children	Yes, have children living at home sometimes	Yes, have children living at home	34%	32%
Age	40~44	40~44	24%	31%
Job	Property / Real Estate	Healthcare/ Medical	12%	31%

6 Conclusion and Further Work

Recommendations for users in social network sites should reflect both "*interests of senders*" and "*interest of recipients in the sender*". We propose a new method that combines these two interests using a weighted harmonic mean. Various combinations

of weights for these two interests were examined in the experiments. Experimental results show our method obtains significantly higher success rate, up to 36% higher than the base line success rate of 16.7%. We also observed that the success rates differs significantly for different types of users (see Table 5).

As this experiment is a retrospective study, we could only measure success rates on the historical interactions that were initiated by senders. This also means that such interactions incorporate senders' interest, because recipients were selected by senders. This also explains why overall success rates for different weight values are not very different. However, in future we plan to study how a sender's activity is influenced by our recommendations. In particular, how users respond to recommendations generated by different weight values.

In future, we also plan to investigate alternative rule generation methods. For example, relaxing a rule by relaxing possible values an attribute could be assigned to. For example, location= "**Sydney City**" could be relaxed by location= "**Sydney City**" OR "**Sydney North**". This would allow us to generalise a rule using small incremental steps. We could also use domain knowledge in selecting or rejecting certain attributes in the process of rule creation. For example, we could consider attributes like location, age, etc to be very important and therefore they should be included in a final rule. We could also consider using different weights for calculating weighted harmonic-mean values for different types of users based on their past activities.

Currently we do not rank recommendations generated by a rule. In future, we plan to rank recipients, for a given rule, based on their likelihood of replying positively to a given sender. This could be very useful if only a small number of recommendations are made. We would also like to explore alternatives to our simplifying assumption that each attribute can be considered separately in calculating preferences between senders and recipients.

Acknowledgments. This research has been supported by the Smart Services Cooperative Research Centre.

References

1. Pazzani, M.J., Billsus, D.: Content-Based Recommendation Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007)
2. Balabanović, M., Shoham, Y.: Fab: Content-Based, Collaborative Recommendation. Communications of the ACM 40(3), 66–72 (1997)
3. Linden, G., Smith, B., York, J.: Amazon.Com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing 7(1), 76–80 (2003)
4. Deshpande, M., Karypis, G.: Item-Based Top-N Recommendation Algorithms. ACM Transactions on Information Systems (TOIS) 22(1), 143–177 (2004)
5. Karypis, G.: Evaluation of Item-Based Top-N Recommendation Algorithms. In: Tenth International Conference on Information and Knowledge Management, pp. 247–254 (2001)
6. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-Based Collaborative Filtering Recommendation Algorithms. In: 10th International Conference on World Wide Web, pp. 285–295 (2001)

7. Huang, Y., Contractor, N., Yao, Y.: Ci-Know: Recommendation Based on Social Networks. In: 9th Annual International Digital Government Research Conference, pp. 375–376 (2008)
8. Mobasher, B., Burke, R., Bhaumik, R.: Attacks and Remedies in Collaborative Recommendation. IEEE Intelligent Systems 22(3), 56–63 (2007)
9. Palau, J., Montaner, M., de la Rosa, J.L.: Collaboration Analysis in Recommender Systems Using Social Networks. In: The Third International Joint Conference on Autonomous Agents and Multi Agent Systems, pp. 137–151 (2004)
10. Perugini, S., Gonçalves, M.A., Fox, E.A.: Recommender Systems Research: A Connection-Centric Survey. Journal of Intelligent Information Systems 23(2), 107–143 (2004)
11. Figueira, J.R., Greco, S., Ehrgott, M.: Multiple Criteria Decision Analysis: State of the Art Surveys. International Series in Operations Research & Management Science, vol. 78 (2005)
12. Adomavicius, G., Kwon, Y.: New Recommendation Techniques for Multicriteria Rating Systems. IEEE Intelligent Systems 22(3), 48–55 (2007)
13. Lee, H.-H., Teng, W.-G.: Incorporating Multi-Criteria Ratings in Recommendation Systems. In: IEEE International Conference on Information Reuse and Integration (IRI 2007), pp. 273–278 (2007)
14. Rattanajitbanjong, N., Maneeroj, S.: Multi Criteria Pseudo Rating and Multidimensional User Profile for Movie Recommender System. In: 2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2009, pp. 596–601 (2009)

Visualising Intellectual Structure of Ubiquitous Computing

Maria R. Lee[1] and Tsung Teng Chen[2]

[1] Shih Chien University, Taiwan
maria.lee@mail.usc.edu.tw
[2] National Taipei University, Taiwan
misttc@mail.ntpu.edu.tw

Abstract. It is difficult to reveal the growth of scientific knowledge even in one's own specialise field due to the enormous amount of research publications available. Providing scientists with knowledge visualisation tools to discover the existence of a scientific paradigm and movements of such paradigms is a challenge task. This paper introduces the state of the art of visualising knowledge structures. The aim of visualising knowledge structures is to capture intellectual structures of a particular knowledge domain. Approaches to the visualisation of knowledge structures with emphasis on the role of citation-based methods are described. Literature published in the online citation data bases CiteSeer and Web of Science (WoS) are exploited to drive the main research themes and their inter-relationships in ubiquitous computing. The Pearson correlation coefficients between items (papers) were used as the basis for PFNET scaling. The intellectual structure map of ubiquitous computing can be revealed by applying PFNET to the main research themes. The major research themes and their interrelationships can be easily identified by the intellectual structure map. The benefit of the results obtained could be for someone new to a specific domain in research study.

Keywords: Knowledge Visualisation, Ubiquitous Computing, Factor Analysis, Pathfinder Network, Intellectual Structure Map.

1 Introduction

Computing technology is a paradigm shift where technology becomes virtually invisible in our lives and is a rapidly advancing and expanding research and development field in this decade. It is difficult to reveal the growth of scientific knowledge even in one's own specialise field due to the enormous amount of research publications available. Providing scientists with knowledge visualisation tools to discover the existence of a scientific paradigm and movements of such paradigms is a challenge task. The main scientific research themes are also very difficult to analyze and grasp by using the traditional methodologies. For example, visualising intrinsic structures among documents in scientific literatures could only capture some aspects of scientific knowledge.

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 261–272, 2010.

This paper introduces the state of the art of visualising knowledge structures. The aim of visualising knowledge structures is to capture and reveal insightful patterns of intellectual structures shared by scientists in a subject field. This paper describes approaches to the visualisation of knowledge structure with emphasis on the role of citation-based methods. Instead of depending upon occurrence patterns of content-bearing words, we aim to capture the intellectual structures of a particular knowledge domain.

We focus on the study of ubiquitous computing, also called pervasive computing. Numerous journals and conferences are now dedicated to the study of ubiquitous computing and related topics. Ubiquitous computing was first described by Weiser [1]. Since then, a rich amount of related literatures are published. It would be useful if the content of those publications could be summarised and presented in an easy way to capture structures and facilitate the understanding of the research themes and trend in ubiquitous computing. The Pearson correlation coefficients between items (papers) were used as the basis for PFNET scaling. The intellectual structure map of ubiquitous computing can be revealed by applying PFNET to the main research themes. The major research themes and their interrelationships can be easily identified by the intellectual structure map.

The goal of the paper is to show the scope and main themes of ubiquitous computing research. We begin by examining the survey studies of visualising knowledge structures. Next, the data collection method and intellectual structure techniques, factor analysis, Pathfinder Network, and intellectual structural map are introduced. The results of the analysis are presented and discussed.

2 Visualising Knowledge Structures

Information visualisation techniques have become a rapid growth research area since the last decade [2-3]. In information retrieval, the vector-space model [4] is an originally and popularly exploited framework for indexing documents based on term frequencies. A focal part of modern statistical probability modelling is Bayesian theorem [5], which focuses on the probabilistic relationship between multiple variables and determining the impact of one variable on another. Shannon's information theory [6] describes information could be treated as a quantifiable value in communications. Self-organised feature maps are essentially classification processes through a neural network. Lin et al. [7] is the first to use self-organised maps to visual information retrieval. WEBSOM organizes textual documents for exploration and search based on self-organised map [8].

Visualising knowledge structure is an art of making maps, which shares some intrinsic characteristics with cartography [9]. Number of useful knowledge visualisation techniques has been applied to detect and extract significant elements from unstructured text. The basis for the visualisation of knowledge structures is formed by the interrelationships between these elements. Citation indexing has been widely applied since 1950s. One of the fundamental objectives of science mapping is to identify the trend associated with a field of study. The map created through citation analysis provides a series of historical data, which cover the literature year by year [10]. These maps show intrinsic semantic connections among disciplines of domains. The author co-citation analysis (ACA) was introduced to discover how scientists in a particular

subject field are intellectually interrelated as perceived by authors in their scientific publications [11]. An intellectual structure of prominent authors in the field provides a respectable source for knowledge visualisation.

Knowledge Domain Visualisation (KDV) depicts the structure and evolution of scientific fields [12]. Some recent works in knowledge discovery and data mining systems compose analysis of engineering domain [13-14].

2.1 Factor Analysis

Factor analysis is one of the commonly used methods in author co-citation analysis. It has been used to identify the intrinsic dimensionality of given co-citation data in a subject domain. White and McCain [15] demonstrate the author co-citation analysis of the information science field that some authors indeed belong to several specialties simultaneously. However, if datasets is big, then the size of the corresponding author co-citation matrix could be large and the analysis becomes computationally complicate and expensive.

White and McCain introduces the raw co-citation should be transformed into Pearson's correlation coefficients using the factor analysis [15]. The correlation coefficients measure the nearness between authors' co-citation profiles. Principal component analysis (PCA) is a suggested alternative to extract factors. The default criterion, Eigen values greater than 1, is normally chosen to decide the number of factors extracted. Missing data should also be replaced by mean co-citation counts for corresponding authors. Pearson's correlation coefficient can be used as a measure of similarity between pairs of authors.

2.2 Pathfinder Network Scaling

Pathfinder network scaling is originally developed by cognitive psychologists for structuring modelling [16]. Pathfinder network scaling relies on the triangle inequality condition to select the most salient relations from proximity data. The Pathfinder network (PFNET), the results of Pathfinder network scaling, consists of all the vertices from the original graph. The number of edges in a Pathfinder network is driven by the basic structure of semantics. The topology of a PFNET is decided by two parameters q and r. The corresponding network is denoted as PFNET (q, r). The q-parameter controls the scope that the triangular inequality condition should be set. The r-parameter is used to computing the distance of a path. The weight of a path with k links is determined by weights $w1, w2, \ldots, wk$ of each individual link as follows.

$$W(P) = \left[\sum_{i=1}^{k} w_i^r \right]^{\frac{1}{r}}$$

3 Intellectual Structure of Ubiquitous Computing

Numerous amounts of scientific papers publish every year and the accumulated literatures over the years are voluminous. We utilized the methods that have been

developed in visualizing information structure to comprehend the entire body of scientific knowledge. The aim is to discover the development in ubiquitous computing.

3.1 Intellectual Structure Process

Figure 1 shows a proposed intellectual structure process to construct a full citation graph from the data drawn from online citation databases, WOS and CiteSeer. The proposed procedure leverages the citation index by using key phrases "ubiquitous computing" to query the index and retrieve all matching documents from the database. The documents retrieved by the query are then used as the initial seed set to retrieve papers that are citing or cited by literatures in the initial seed set [17-19]. The co-citation matrix is derived from the co-citation relationships between papers. A co-citation relationship existed between two papers when a third paper cites them both, i.e., both papers are listed in the reference portion of the third paper. The full citation graph is built by linking all articles retrieved, which includes more documents than the other schemes reviewed earlier.

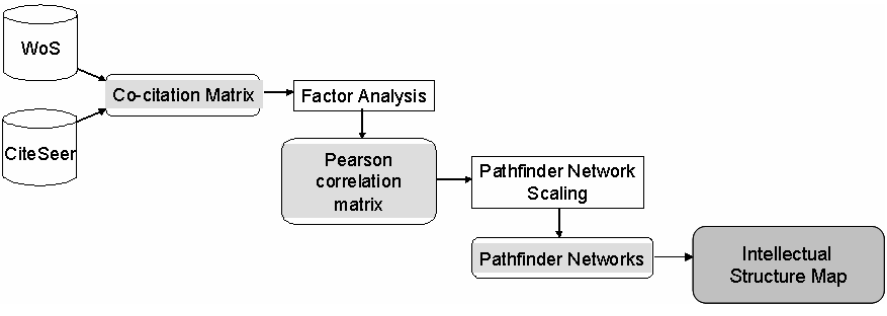


Fig. 1. Intellectual Structure Process

The main usage of the factor analysis is to reduce the number of variables and to detect structure in the relationships between variables. Factor analysis combines correlated variables (papers) into one component (research theme). The co-citation matrix is the input of factor analysis. The co-citation graph is represented by a matrix to compute the correlation matrix of Pearson's correlation coefficients. The Pearson correlation matrix, which is resulted from the factor analysis, is the input of the Pathfinder network scaling. All nodes in a graph are connected by weighted links. The weights are represented by the value of correlation coefficients for each pair of documents.

The citation data are driven from two online citation databases, Citeseer and IS Web of Science (WoS). CiteSeer is an open access free database, which is a scientific literature digital library. Citseer's search engine focuses primarily on the literature in computer and information science. ISI Web of Knowledge database is created by Thomson Reuters in 1997 and integrated access to high quality, multidisciplinary research literature. Web of Science (WoS) is part of ISI Web of Knowledge. WoS covers SCI, SSCI and A&HCI citation databases.

Citation data are collected by querying both databases with the key phrases "ubiquitous computing" and retrieving the initial key papers' information. The key papers are then used as the initial seed set to retrieve papers that are citing or are cited by literatures in the initial seed set [20]. A full citation graph is generated by linking all articles retrieved. The depth of the expanded search is restricted to three layers to maintain the most relevant literatures.

3.2 Main Components Derived from Factor Analysis

Factor analysis is applied as a data reduction and structure detection method. The co-citation matrixes generated from CiteSeet and WoS are derived from the citation graphs and fed to factor analysis. The unit of analysis is based on documents rather than author due to a researcher's specialty may evolve over time [18,21].

49 components with Eigenvalue over one were identified from the CiteSeer citation data. These factors collectively explained approximately 84.2% total variances. Papers with a loading over 0.6 to a component are collected and studies to determine the content of the component. A proper descriptive name for each component is decided, which represent the research trends in the ubiquitous computing field. Based on CiteSeer citation data, top 8 components and the variances of the components explained are listed in Table 1. Component 9 does not include any papers with loading larger than 0.6 and therefore, is not listed in the table. The content of these nine factors are described in the context of ubiquitous computing.

Table 1. Top 8 Factors of ubiquitous computing drew from CiteSeer

Factor	Component Name	Variance Explained	Description
1	Routing protocols for mobile and ad hoc networks	7.206	An ad hoc network is a collection of wireless mobile nodes dynamically forming a temporary network without the use of any existing network infrastructure or centralized administration.
2	Location and data management in mobile wireless environment	6.733	Location and data management in a mobile wireless environment is different from the transitional fixed wire environment. Location and data management is required when all the communications over a mobile wireless environment with a sufficient and steady bandwidth.
3	Location and context aware computing	5.702	Location and context aware computing will free the user from the traditional constraints of the desktop. Context refers to the physical and social situation in which computational devices are embedded. Contextual information can be used to provide services that are appropriate to the situational events.

Table 1. (*continued*)

4	Broadcast based data management for asymmetric communication environments	5.023	Broadcasted data has been proposed as a means to deliver data efficiently to clients in asymmetric environments, where the available bandwidth from the server to the clients exceeds the bandwidth in the opposite direction. In the presence of such asymmetry, applications must rely on the broadcast data channel to receive the up-to-date information.
5	Integrating with computer augmented artifacts and environment	4.310	Interacting with computer augmented artifacts and environments may greatly enhance a user's experience. Computer augmented physical objectives or devices may facilitate more effectively computational mediation.
6	Transmission control protocol (TCP) over mobile internetworks	4.120	The study of TCP over mobile internetworks addresses the performance issue of reliable data communication in mobile computing environments. Two changed assumptions need to be addressed in the mobile computing: (1) the end points of the communication link are fixed and (2) the underlying network has a high and reliable bandwidth with low latency.
7	Application design for mobile computing	3.524	This factor addresses the disparity of mobile devices in resources, network characteristics, display size, and method of input from the application level. Application design strategies may reduce the demands placed on the wireless network.
8	Disconnected operations	3.023	Disconnected operation is a mode of operation that enables a client to continue accessing critical data during temporary failures of a shared data repository. The temporary failures may due to networks or data sources breakdown. The core idea behind this work is utilizing tradition performance improving data, such as caching data, to improve availability.

From the WoS dataset, 30 components with Eigenvalue over one collectively explained approximately 86.4% total variances. These components are selected as the representative major themes of ubiquitous computing. Papers with a loading over 0.5 to a component are collected and studied to determine the content of the component. Based on WOS citation data, top 8 components with descriptions and the variances of the components explained are listed in Table 2.

Table 2. Top 8 Factors of ubiquitous computing drew from Web of Science (WOS)

Factor	Component Description	Variance Explained	Description
1	Foundational studies of ubiquitous computing	17.123	Foundational studies of ubiquitous computing provide a generic platform for location and spatial-aware systems. The platform supports a unified spatial-aware infrastructure based on digital models of the physical world. A universal spatial and context-aware infrastructure is essential to overcome the sheer diversity of exploitable contexts and the myriad of sensing technologies.
2	Power aware routing protocol for wireless sensor network	8.329	The availability of small, lightweight low-cost network is crucial to the success of ubiquitous computing. The lightweight network uses energy sparingly to prolong the operational span of the ubiquitous network. The power saving algorithms and protocols are the focus of much ubiquitous computing related research.
3	Medical informatics, application of ubiquitous computing in health care	6.917	The ubiquitous availability of clinical information is major trend in medical informatics research. The application of new information and communication technologies will offer new opportunities and increase the potential of medical informatics methods and tools. The mobility of hospital environment, such as staff, patients, documents and equipments, makes hospitals' ideal applications for pervasive or ubiquitous computing technology.
4	Context-aware workflow language based on Web services	5.105	Research in this factor seems to explore the common feature of Web services and ubiquitous computing. According to W3C, the web services are defined as "a software system designed to support an interoperable machine to machine interaction over a network". The standardization of ubiquitously available services and interoperability between services (factor 1) becomes the natural bond between web services and ubiquitous computing.
5	Context-aware computing	4.878	Papers in factor 5 try to clarify and define the scope and content of context aware computing. Context awareness is the key to dispersing and enmeshing ubiquitous computation in our lives. Contextual information is acquired and utilized by devices to provide services that are appropriate to the situational events.

Table 2. (*continued*)

6	Ambient intelligent systems	4.854	Papers in factor 6 extended the context-aware computing to ambient intelligent systems. How "context" can be effectively utilized by a context-aware system, which in turn exhibit ambient intelligent is the focal issue of research in this factor.
7	Open services gateway initiative (OSGi)	4.766	OSGi is a technology standard that can coordinate diverse device technologies and enable compound services across different networking technologies. OSGi can be viewed as an initial effort of commercial realisation of the universal spatial and context-aware infrastructure envisioned by the academy.
8	Ubiquitous applications in education	4.350	Articles in factor 8 explore how learning could be augmented by ubiquitous computing devices in an educational setting. The functionalities of traditional classroom equipment and instruments such as whiteboard, notebook PC, PDAs and other learning aids could be augmented by embedding ubiquitous computing power to enhance the learning environment and enriching the learning experiences.

3.3 Pathfinder Network

The Pearson's correlation coefficients between items (papers) are calculated and used as the basis for PFNET scaling. The value of the Pearson's correlation coefficient falls between the range -1 and 1. Highly correlated items are placed closely together spatially. The nodes located close to the centre of a PFNET graph represents papers contributed to a fundamental concept, which are frequently referred by other peripheral literature that are positioned in outer branches. The distance between items is inversely propositional to the correlation coefficient, which maps less correlated items apart and highly correlated items spatially adjacent.

Left-hand site of figure 2 represent PFNET scaling of ubiquitous computing from CiteSeer. Articles under the same factor are painted with the same colour. The number in the parenthesis is the factor number which an article belongs to. Cyan nodes with (0) represent articles that are not assigned to any factor. The top ranked components cluster numbered 6, 7, and 12 locate closely to the centre of the PFNET graph (surrounded by a big circle in the centre of the graph), which suggests papers in these components play fundamental roles in ubiquitous computing. A foundational but low-ranked factor may be interpreted as an underdeveloped but important theme in the research field. Component 4 (top left circle, node colour in blue), although ranked high in the amount of variance explained, plays peripheral roles in ubiquitous computing related research. A high-ranked peripheral component is generally an important

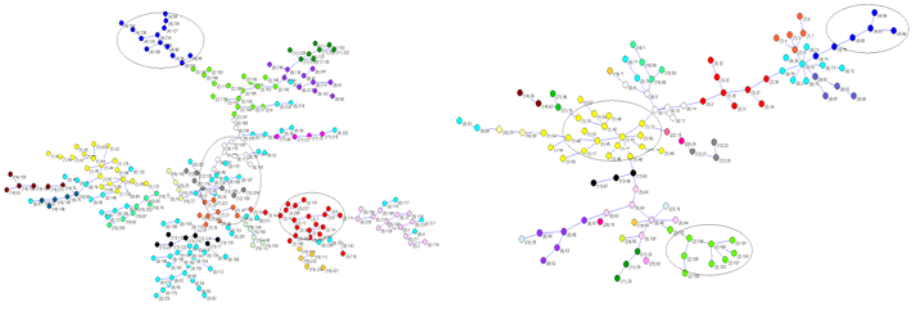


Fig. 2. PFNET Scaling of ubiquitous computing drew from CiteSeer and WoS.

study, but plays only a supplement role in ubiquitous computing study. Component 3 (lower right circle, node colour in red) plays an interesting role, which indicates that location and context aware computing related researches are important to ubiquitous computing and are important topics in general.

Right-hand site of figure 2 represent PFNET scaling of ubiquitous computing from Web of Science (WoS). Articles under the same factor are painted with the same colour. The number in the parenthesis is the factor number which an article belongs to. Top ranked component cluster number 1, 5, and 6 locate closely to the centre of the PFNET graph (surrounded by a big circle in the centre of the graph), which suggests papers in these components play foundational roles in ubiquitous computing research. Component cluster number 2 (lower-right circle, node colour in green) and 4 (upper right circle, node colour in blue), although ranked high in the amount of variance explained, they play peripheral roles in ubiquitous computing related research. Component cluster number 3 (node colour in red) plays an interesting role, which indicates that medical informatics research is important to ubiquitous computing as well as an important topic in general.

3.4 The Relationships of the Intellectual Structure Map

The Pearson correlation coefficients between items (papers) were used as the basis for PFNET scaling. The nodes located close to the centre of a PFNET graph represent papers that contribute to a core concept. The intellectual structure map of ubiquitous computing can be revealed by applying PFNET to the main research themes. Figure 3 derived from figure 2 by combining nodes in the same factor into a block to highlight the relationships between theses themes. The left-hand side of the figure 3 represents the relationships between the main components in the intellectual structure map drew from CiteSeer whereas the right-hand side of figure 3 drew from WoS. The double-arrowed lines indicate a connection relationship. The most connected components in the intellectual structure map in CiteSeer is application design for mobile computing (7) whereas the most connected components in WoS is context-aware computing (5). The major research themes and their interrelationships can be easily identified by the intellectual structure map.

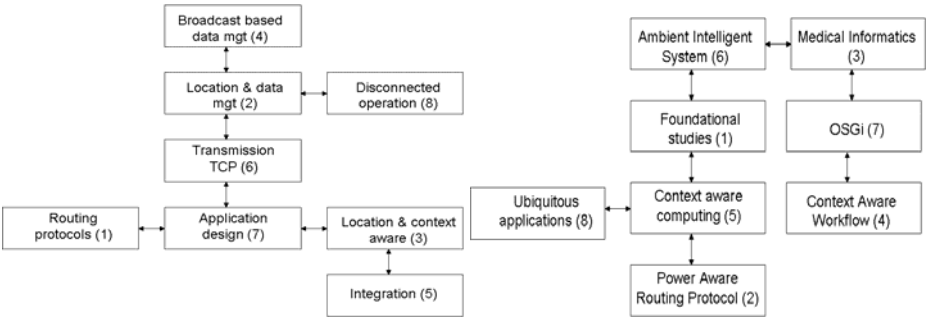


Fig. 3. The intellectual structure map drew from CiteSeer and WoS

4 Discussion

Abowd and Munatt [22] investigate the main research focuses in ubiquitous computing publications in 2000, which include natural interface, context-aware applications and automated capture and access. However, they mainly based on the bounded expertise of the author(s) and a rather limited set of references. We propose a visualising knowledge structure method in analyzing large collections of literatures, which reveals the major research themes and their inter-relationships in ubiquitous computing. We utilize the intellectual structure construction and knowledge domain visualisation techniques developed by the information scientists to ease the task of understanding the main research themes in ubiquitous computing.

Based on the citation papers derived from factor analysis and PFNET in ISI Web of Science (WoS) in 2008, foundational ubiquitous computing studies, context-aware computing, and ambient intelligent systems provide a fundamental and important knowledge base to ubiquitous computing studies. Power aware routing protocol and context-aware workflow language are relevant and important studies in general, but only play a supporting or supplemental role in ubiquitous computing research.

In contrast, based on the citation data drew from CiteSeer in 2008, the study of application design for mobile computing, TCP over mobile internetworks and network support for real-time applications provide a fundamental and important technical knowledge base to ubiquitous computing studies. Broadcast based data management for asymmetric communication environments and interacting with computer augmented artefacts and environment are relevant and important studies in general, but only play a supporting or supplemental role in ubiquitous computing studies.

The difference between main themes drew from CiteSeer and WoS CiteSeer is due to that CiteSeer citation index is primarily a computer, information science and engineering citation database, whereas WoS is a comprehensive index. The intellectual structure derived from a predominantly science and engineering oriented index is biased toward the technical aspect of ubiquitous computing. In contrast, WoS is a comprehensive citation database. WoS reveals the application and business themes as well as the technical one. The most connected components in the intellectual structure map in CiteSeer is application design for mobile computing (7) whereas the most

connected components in WoS is context-aware computing (5). The major research themes and their interrelationships can be easily identified by the intellectual structure map.

5 Conclusion

Providing scientists with knowledge visualization tools to reveal the scientific paradigm and movements of such a paradigm is a challenge task. We have introduced the method of visualizing knowledge structures with emphasis on the role of citation-based methods. Factor analysis and Pathfinder Network are used to discover new and significant developments of intellectual structure in the ubiquitous computing research field. The Pearson correlation coefficients between items (papers) were used as the basis for PFNET scaling. The intellectual structure map of ubiquitous computing can be revealed by applying PFNET to the main research theme. Literature published in the online citation databases CiteSeer and Web of Science (WoS) in 2008 were explored to drive the research themes.

We tried to provide a broader view of ubiquitous computing study by applying intellectual structure methods developed by information scientists. The main themes can be uncovered with respect to fundamental and important knowledge as well as supporting or supplemental knowledge in the ubiquitous computing domain. The results obtained show that the study of application design for mobile computing, TCP over mobile internetworks, network support for real-time applications, foundational ubiquitous computing studies, context-aware computing, and ambient intelligent systems are fundamental topics in ubiquitous computing. Broadcast based data management for asymmetric communication environments, interacting with computer-augmented artefacts and environment, Power aware routing protocol and context-aware workflow language are relevant and important studies in general, but only play a supporting or supplemental role in ubiquitous computing research.

The benefit of the results obtained could be for someone new to a specific domain in research study. The major research themes and their interrelationships can be easily identified by the intellectual structure map. The proposed method may be re-used in other disciplines and share across different research domains. One of the future directions is to apply this proposed method to leverage the research theme networks, which is intellectually interrelated the relationships among publications, citations, research projects, and even patents. We also plan to explore further the interdisciplinary researches in future studies.

References

1. Weiser, M.: The computer for the 21st century, vol. 265, pp. 66–75. Scientific American (1991)
2. Card, S., Mackinlay, J., Shneiderman, B.: Readings in Information Visualisation: Using Vision to Think. Morgan Kaufmann, San Francisco (1999)
3. Chen, C.: Information Visualisation and Virtual Environments. Springer, London (1999)
4. Salton, G.: Developments in automatic text retrieval. Science 253, 974–980 (1991)
5. Neal, R.: Bayesian Learning for Neural Networks. Springer, New York (1996)

6. Shannon, C.: A mathematical theory of communication. Bell System Technical Journal 27, 379–423, 623–656 (1948)
7. Lin, X., Soergel, D., Marchionini, G.: A self-organising semantic map for information retrieval. In: SIGIR 1991, pp. 262–269. ACM Press, New York (1991)
8. Lagus, K., Honkela, T., Kaski, S., Kohonen, T.: WEBSOM - A Status Report. In: Proceedings of STeP 1996, pp. 73–78. Publications of the Finnish Artificial Intelligence Society, Jarmo Alander (1996)
9. Chen, C.: Visualisation of Knowledge Structures. The Handbook of Software Engineering and Knowledge Engineering 2, 700–744 (2002)
10. Garfield, E.: Citation indexes for science: a new dimension in documentation through association of ideas. Science 122, 108–111 (1975)
11. White, H., Griffith, B.: Author co-citation: A Literature measure of intellectual structure. Journal of the American Society for Information Science 32, 163–172 (1981)
12. Borner, K., Chen, C., Boyack, K.: Visualising Knowledge Domains. Annual Review of Information Science and Technology 37, 179–255 (2002)
13. Mothe, J., Dousset, B.: Mining document contents in order to analyse a scientific domain. In: Sixth International Conference on Social Science Methodology (2004)
14. Mothe, J., Chrisment, C., Dkaki, T., Dousset, B., Karouach, S.: Combining mining and visualisation tools to discover the geographic structure of a domain. Computers, Environment and Urban Systems 30, 460–484 (2006)
15. White, H., McCain, K.: Visualising a discipline: an author co-citation analysis of information science. Journal of American Society for Information Science 49(4), 327–356 (1995)
16. Schcaneveldt, R., Durso, F., Dearholt, D.: Network structures in proximity data. In: The Psychology of Learning and Motivation, vol. 24, pp. 249–284. Academic Press, London (1989)
17. Lee, M., Chen, T.: Visualizing Trends in Knowledge Management. In: Zhang, Z., Siekmann, J.H. (eds.) KSEM 2007. LNCS (LNAI), vol. 4798, pp. 362–371. Springer, Heidelberg (2007)
18. Chen, T., Lee, M.: Revealing Themes and Trends in the Knowledge Domain's Intellectual Structure. In: Hoffmann, A., Kang, B.-h., Richards, D., Tsumoto, S. (eds.) PKAW 2006. LNCS (LNAI), vol. 4303, pp. 99–107. Springer, Heidelberg (2006)
19. Pozi, L.: The Intellectual Structure and Interdisciplinary Breath of Knowledge Management: a Bolometric Study of Its Early Stage of Development. Scientometrics 55, 259–272 (2002)
20. Chen, T., Xie, L.: Identifying Critical Focuses in Research Domains. In: Proceedings of the Information Visualization, Ninth International Conference on (IV 2005), pp. 135–142 (2005)
21. Lee, M., Chen, T.: From Knowledge Visualization Techniques to Trends in Ubiquitous Multimedia Computing. In: 2008 International Symposium on Ubiquitous Multimedia Computing, pp. 73–78 (2008)
22. Abowd, G., Munatt, E.: Charting past, present and future research in ubiquitous computing. ACM Transactions on Computer-Human Interaction 17, 29–58 (2000)

Acquiring Expected Influence Curve from Single Diffusion Sequence

Yuya Yoshikawa[1], Kazumi Saito[1], Hiroshi Motoda[2], Kouzou Ohara[3], and Masahiro Kimura[4]

[1] School of Administration and Informatics, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
{b7101,k-saito}@u-shizuoka-ken.ac.jp
[2] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp
[3] Department of Integrated Information Technology, Aoyama Gakuin University
Kanagawa 229-8558, Japan
ohara@it.aoyama.ac.jp
[4] Department of Electronics and Informatics, Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp

Abstract. We address the problem of estimating the expected influence curves with good accuracy from a single observed information diffusion sequence, for both the asynchronous independent cascade (AsIC) model and the asynchronous linear threshold (AsLT) model. We solve this problem by first learning the model parameters and then estimating the influence curve using the learned model. Since the length of the observed diffusion sequence may vary from a very long one to a very short one, we evaluate the proposed method by simulation using artificial diffusion sequence of various lengths and show that the proposed method can estimate the expected influence curve robustly from a single diffusion sequence with various lengths.

1 Introduction

The rise of the Internet and the World Wide Web accelerates the creation of various large-scale social networks, and considerable attention has been brought to social networks as an important medium for the spread of information [1,2,3,4,5]. Innovation, topics and even malicious rumors can diffuse through social networks in the form of so-called "word-of-mouth" communications. Such social interaction processes are usually characterized by highly distributed phenomena over a social network, but the complexity and distributed nature of these processes do not necessarily imply that these evolutions are chaotic or unpredictable. Just as natural scientists discover laws and create models for their fields, so can one, in principle, find empirical regularities and develop explanatory accounts of evolution in a social network. Especially, such predictive knowledge would be valuable for market opportunities. In this paper, as a piece of such predictive knowledge, we focus on acquiring the expected influence curve of each information source node by using information diffusion models.

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 273–287, 2010.

Widely used information diffusion models in recent studies are the *independent cascade (IC)* [6,7,8] and the *linear threshold (LT)* [9,10]. They have been used to solve such problems as the *influence maximization problem* [7,11]. These two models focus on different information diffusion aspects. The IC model is sender-centered and an active node influences its inactive neighbors *independently* with diffusion probabilities assigned to links. On the other hand, the LT model is receiver-centered and a node is influenced by its active neighbors if the sum of their weights exceeds the threshold for the node. Both models have parameters that need be specified in advance: diffusion probabilities for the IC model, and weights for the LT model. However, their true values are not known in practice. This poses yet another problem of estimating them from a set of information diffusion results that are observed as time-sequences of influenced (activated) nodes. To the best of our knowledge, there are only a few methods that can estimate the parameter values for the IC and LT models and their variants that incorporate asynchrnous time delay (referred to as the AsIC model and the AsLT model) [3,12,13,14]. We follow the methods in [13,14] in this paper.

Now assume that we observed a single information diffusion sequence for an information source node. How can we acquire the expected influence curve from this single instance of observation? This is the problem we want to solve. In a sence, this sequence can be regarded as a piece of crude knowledge about the expected influence curve because we can count the number of nodes that have been influenced (activated) by any time point t which we specify. However, due to its stochastic nature, such a sequence varies in a quite wide range each time we observe it, even if we know which of the two models (AsIC and AsLT) the information diffusion follows. Thus, it is undesirable to approximate the expected influence curve by a single instance of observed sequence.

In this paper, we assume that information diffuses over a network by either the AsIC model or the AsLT model, and propose a novel method for estimating the expected influence curve by first estimating parameters for the assumed models from a single observed informaition diffusion sequence and use the learned model to estimate the expected curve. In another word, our method can be viewed as a knowledge refinement method from the observed single information diffusion sequence to the expected influence curve based on the information diffusion model. We performed extensive experiments to evaluate whether the proposed method can estimate the influence curve much more accurately than the observed diffusion curve itself. The results cleary show the advantage of our method.

The paper is organized as follows. We revisit the information diffusion models and briefly explain the independent cascade model, the linear threshold model, and their asynchronous time delay versions (the models we use in this paper) : AsIC and AsLT in section 2, and revisit parameter learning algorithms for AsIC and AsLT in section 3. We then describe the estimation method of the expected influence curve in section 4, and explain the experimental results in detail in section 5, followed by some discussions in section 6. We summarize our conlcusion in section 7.

2 Information Diffusion Models

We first define the IC model according to [7], and then introduce the asynchronous IC model (AsIC). After that, we do the same for the LT model and the asynchronous

LT model (AsLT). We mathematically model the spread of information over a directed network $G = (V, E)$ without self-links, where V and E ($\subset V \times V$) stands for the sets of all the nodes and links, respectively. We call nodes *active* if they have been influenced with the information. It is assumed that nodes can switch their states only from inactive to active, but not from active to inactive. Given an initial set S of active nodes, we assume that the nodes in S have first become active at an initial time, and all the other nodes are inactive at that time. Node u is called a *child node* of node v if $(v, u) \in E$, and node u is called a *parent node* of node v if $(u, v) \in E$. For each node $v \in V$, let $F(v)$ and $B(v)$ denote the set of child nodes of v and the set of parent nodes of v, respectively,

$$F(v) = \{w \in V;\ (v, w) \in E\}, \quad B(v) = \{u \in V;\ (u, v) \in E\}.$$

2.1 Independent Cascade Model

The IC model is a fundamental probabilistic model for the spread of a disease. In this model, we specify a real value $\kappa_{u,v}$ with $0 < \kappa_{u,v} < 1$ for each link (u, v) in advance. Here $\kappa_{u,v}$ is referred to as the *diffusion probability* through link (u, v). The diffusion process unfolds in discrete time-steps $t \geq 0$, and proceeds from a given information source node in the following way. When a node u becomes active at time-step t, it is given a single chance to activate each currently inactive child node v, and succeeds with probability $\kappa_{u,v}$. If u succeeds, then v will become active at time-step $t + 1$. If multiple parent nodes of v become active at time-step t, then their activation attempts are sequenced in an arbitrary order, but all performed at time-step t. Whether or not u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

2.2 Asynchronous Independent Cascade Model

We extend the IC model so as to allow continuous-time delays, and refer to the extended model as the *Asynchronous independent cascade (AsIC) model*. In the AsIC model, we specify a real value $r_{u,v}$ with $r_{u,v} > 0$ for each link $(u, v) \in E$ in advance together with diffusion parameter $\kappa_{u,v}$. We refer to $r_{u,v}$ as the *time-delay parameter* through link (u, v).

The diffusion process unfolds in continuous-time t, and proceeds from a given information source node in the following way. Suppose that a node u becomes active at time t. Then, node u is given a single chance to activate each currently inactive child node v. We choose a delay-time δ from the exponential distribution with parameter $r_{u,v}$. If node v is not active before time $t + \delta$, then node u attempts to activate node v, and succeeds with probability $\kappa_{u,v}$. If u succeeds, then v will become active at time $t + \delta$. Under the continuous time framework, it is unlikely that multiple parent nodes of v attempt to activate v at exactly the same time $t + \delta$. So we ignore this possibility. Whether or not u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

For an information source node v, let $\varphi(t; v)$ denote the number of active nodes at a specified time t, i.e. the number of nodes that have become activated by t. Note that $\varphi(t; v)$ is a random variable. Let $\sigma(t; v)$ denote the expected value of $\varphi(t; v)$. We call $\sigma(t; v)$ the *expected influence curve* of v for the AsIC model.

2.3 Linear Threshold Model

The LT model is a fundamental probabilistic model for the spread of innovation. In this model we specify a *weight* ($\omega_{u,v} > 0$) for every node $v \in V$ from its parent node u in advance such that $\sum_{u \in B(v)} \omega_{u,v} \leq 1$. The diffusion process from a given initial active set S proceeds according to the following randomized rule. First, for any node $v \in V$, a *threshold* θ_v is chosen uniformly at random from the interval $[0, 1]$. At time-step t, an inactive node v is influenced by each of its active parent nodes, u, according to weight $\omega_{u,v}$. If the total weight from active parent nodes of v is no less than threshold θ_v, that is, $\sum_{u \in B_t(v)} \omega_{u,v} \geq \theta_v$, then v will become active at time-step $t + 1$. Here, $B_t(v)$ stands for the set of all the parent nodes of v that are active at time-step t. The process terminates if no more activations are possible.

2.4 Asynchronous Linear Threshold Model

We make a similar extention to the LT model so as to allow continuous-time delays, and refer to the extended model as the *Asynchronous linear threshold (AsLT) model*. In the AsLT model, in addition to the weight set $\{\omega_{u,v}\}$, we specify real values r_v with $r_v > 0$ in advance for each node $v \in V$. We refer to r_v as the *time-delay parameter* on node v. Note that r_v depends only on v, which means that it is the node v's decision when to receive the information once the activation condition has been satisfied.

The diffusion process unfolds in continuous-time t, and proceeds from a given initial active set S in the following way. Suppose that the total weight from active parent nodes of v became no less than the threshold θ_v at time t for the first time. Then, v will become active at time $t + \delta$, where we choose a delay-time δ from the exponential distribution with parameter r_v. Further, note that even though some other non-active parent nodes of v become active during the time period between t and $t + \delta$, the activation time of v, $t + \delta$, still remains the same. The other diffusion mechanisms are the same as the LT model. Similarly to the AsIC model, we can also define the expected influence curve $\sigma(t; v)$ of an information source node v for the AsIC model.

3 Learning Algorithms

We define the time-delay parameter vector $\boldsymbol{r}$ and the diffusion parameter vector $\boldsymbol{\kappa}$ by $\boldsymbol{r} = (r_{u,v})_{(u,v)\in E}$ and $\boldsymbol{\kappa} = (\kappa_{u,v})_{(u,v)\in E}$ for the AsIC model. Similarly, we define the parameter vectors $\boldsymbol{\omega}$ and $\boldsymbol{r}$ by $\boldsymbol{\omega} = (\omega_{u,v})_{(u,v)\in E}$ and $\boldsymbol{r} = (r_v)_{v\in V}$ for the AsLT model. In practice, the true values of these parameters are not available. Thus, we must learn them from past information diffusion histories.

We consider an observed data set of M independent information diffusion results, $\{D_m;\ m = 1, \cdots, M\}$. Here, each D_m is a set of pairs of active nodes and their activation times in the mth information diffusion result, $D_m = \{(u, t_{m,u}),\ (v, t_{m,v}),\ \cdots\}$. For each D_m, we denote the observed initial time by $t_m = \min\{t_{m,v};\ (v, t_{m,v}) \in D_m\}$, and the observed final time by $T_m \geq \max\{t_{m,v};\ (v, t_{m,v}) \in D_m\}$. Note that T_m is not necessarily equal to the final activation time. Hereafter, we express our observation data by $\mathcal{D}_M = \{(D_m, T_m);\ m = 1, \cdots, M\}$. For any $t \in [t_m, T_m]$, we set $C_m(t) = \{v;\ (v, t_{m,v}) \in D_m,\ t_{m,v} <$

$t\}$. Namely, $C_m(t)$ is the set of active nodes before time t in the mth information diffusion result. For convenience sake, we use C_m as referring to the set of all the active nodes in the mth information diffusion result. Moreover, we define a set of non-active nodes with at least one active parent node for each by $\partial C_m = \{v;\ (u,v) \in E,\ u \in C_m,\ v \notin C_m\}$. For each node $v \in C_m \cup \partial C_m$, we define the following subset of parent nodes, each of which has a chance to activate v.

$$\mathcal{B}_{m,v} = \begin{cases} B(v) \cap C_m(t_{m,v}) & \text{if } v \in C_m(t_{m,v}), \\ B(v) \cap C_m & \text{if } v \in \partial C_m. \end{cases}$$

In oder to learn the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ for the AsIC model, and the values of $\boldsymbol{r}$ and $\boldsymbol{\omega}$ for the AsLT model for the given $\mathcal{D}_M$, we adopt the method proposed in [13] and [14], respectively, each of which is only briefly explained here.

3.1 Learning Parameters of AsIC Model

To learn the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$ for the AsIC model, we revisit the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\omega}$ to use as the objective function [13]. First, we consider any node $v \in C_m$ with $t_{m,v} > t_m$ for the mth information diffusion result. Let $\Phi_{m,u,v}$ denote the probability density that a node $u \in B(v) \cap C_m(t_{m,v})$ activates the node v at time $t_{m,v}$, that is,

$$\Phi_{m,u,v} = \kappa_{u,v} r_{u,v} \exp(-r_{u,v}(t_{m,v} - t_{m,u})). \tag{1}$$

Let $\Psi_{m,u,v}$ denote the probability that the node v is not activated from a node $u \in B(v) \cap C_m(t_{m,v})$ during the time-period $[t_{m,u}, t_{m,v}]$, that is,

$$\begin{aligned} \Psi_{m,u,v} &= 1 - \kappa_{u,v} \int_{t_{m,u}}^{t_{m,v}} r_{u,v} \exp(-r_{u,v}(t - t_{m,u}))dt \\ &= \kappa_{u,v} \exp(-r_{u,v}(t_{m,v} - t_{m,u})) + (1 - \kappa_{u,v}). \end{aligned} \tag{2}$$

As explained in 2.2, it is not necessary to consider simultaneous activations by multiple active parents even if $\eta = |B(v) \cap C_m(t_{m,v})| > 1$. Thus, the probability density that the node v is activated at time $t_{m,v}$, denoted by $h^{(IC)}_{m,v}$, can be expressed as

$$\begin{aligned} h^{(IC)}_{m,v} &= \sum_{u \in B(v) \cap C_m(t_{m,v})} \Phi_{m,u,v} \left(\prod_{x \in B(v) \cap C_m(t_{m,v}) \setminus \{u\}} \Psi_{m,x,v} \right) \\ &= \prod_{x \in B(v) \cap C_m(t_{m,v})} \Psi_{m,x,v} \sum_{u \in B(v) \cap C_m(t_{m,v})} \Phi_{m,u,v} (\Psi_{m,u,v})^{-1}. \end{aligned} \tag{3}$$

Note that we are not able to know which node u actually activated the node v. This can be regarded as a hidden structure.

Next, for the mth information diffusion result, we consider any link $(v,w) \in E$ such that $v \in C_m$ and $w \notin C_m$. Let $g^{(IC)}_{m,v,w}$ denote the probability that the node w is not activated by the node v during the observed time period $[t_m, T_m]$. We can easily derive the following equation:

$$g^{(IC)}_{m,v,w} = \kappa_{v,w} \exp(-r_{v,w}(T_m - t_{m,v})) + (1 - \kappa_{v,w}). \tag{4}$$

Therefore, by using equations (3), (4), and the independence properties, we can define the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ by

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M) = \prod_{m=1}^{M} \prod_{v \in C_m} \left(h_{m,v}^{(IC)} \prod_{w \in F(v) \setminus C_m} g_{m,v,w}^{(IC)} \right), \tag{5}$$

Thus, our problem is to obtain the time-delay parameter vector $\boldsymbol{r}$ and the diffusion parameter vector $\boldsymbol{\kappa}$, which together maximize Equation (5). To obtain the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, we can employ a learning method based on the Expectation-Maximization algorithm in order to stably obtain its solutions [13].

3.2 Learning Parameters of AsLT Model

To learn the values of $\boldsymbol{r}$ and $\boldsymbol{\omega}$ from $\mathcal{D}_M$ for the AsLT model, we also revisit the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\omega}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\omega}$ to use as the objective function [14]. For the sake of technical convenience, we introduce a slack weight $\omega_{v,v}$ for each node $v \in V$ such that $\omega_{v,v} + \sum_{u \in B(v)} \omega_{u,v} = 1$. Here note that such a slack weight $\omega_{v,v}$ never contributes to the activation of v and that for each node v, since a threshold θ_v is chosen uniformly at random from the interval $[0, 1]$, we can regard each weight $\omega_{*,v}$ as a multinomial probability.

Suppose that a node v became active at time $t_{m,v}$ for the mth result. Then, we know that the total weight from active parent nodes of v became no less than the threshold θ_v at the time when one of these active parent nodes, $u \in \mathcal{B}_{m,v}$, became first active. However, in case of $|\mathcal{B}_{m,v}| > 1$, there is no way of exactly knowing the actual nodes due to the continuous time-delay. Suppose that a node v was actually activated when a node $\zeta \in \mathcal{B}_{m,v}$ became activated. Then θ_v is between $\sum_{u \in B(v) \cap C_m(t_{m,\zeta})} \omega_{u,v}$ and $\omega_{\zeta,v} + \sum_{u \in B(v) \cap C_m(t_{m,\zeta})} \omega_{u,v}$. Namely, the probability that θ_v is chosen from this range is $\omega_{\zeta,v}$. Here note that such events with respect to different active parent nodes are mutually disjoint. Thus, the probability density that the node v is activated at time $t_{m,v}$, denoted by $h_{m,v}^{(LT)}$, can be expressed as

$$h_{m,v}^{(LT)} = \sum_{u \in \mathcal{B}_{m,v}} \omega_{u,v} r_v \exp(-r_v(t_{m,v} - t_{m,u})). \tag{6}$$

Here we define $h_{m,v}^{(LT)} = 1$ if $t_{m,v} = t_m$.

Next, we consider any node $w \in V$ belonging to $\partial C_m = \{w; (v, w) \in E \wedge v \in C_m(T_m) \wedge w \notin C_m(T_m)\}$ for the mth result. Let $g_{m,v}$ denote the probability that the node v is not activated during the observed time period $[t_m, T_m]$. We can calculate $g_{m,v}$ as follows:

$$\begin{aligned} g_{m,v}^{(LT)} &= 1 - \sum_{u \in \mathcal{B}_{m,v}} \omega_{u,v} \int_{t_{m,u}}^{T_m} r_v \exp(-r_v(t - t_{m,u}))dt \\ &= 1 - \sum_{u \in \mathcal{B}_{m,v}} \omega_{u,v}(1 - \exp(-r_v(T_m - t_{m,v}))) \\ &= \omega_{v,v} + \sum_{u \in B(v) \setminus \mathcal{B}_{m,v}} \omega_{u,v} + \sum_{u \in \mathcal{B}_{m,v}} \omega_{u,v} \exp(-r_v(T_m - t_{m,v})). \end{aligned} \tag{7}$$

Therefore, by using Equations (6) and (7), and the independence properties, we can define the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\omega}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\omega}$ by

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\omega}; \mathcal{D}_M) = \prod_{m=1}^{M} \left(\prod_{v \in C_m} h_{m,v}^{(LT)} \right) \left(\prod_{v \in \partial C_m} g_{m,v}^{(LT)} \right). \tag{8}$$

Thus, our problem is to obtain the time-delay parameter vector $\boldsymbol{r}$ and the diffusion parameter vector $\boldsymbol{\omega}$, which together maximize Equation (8). To obtain the values of $\boldsymbol{r}$ and $\boldsymbol{\omega}$, we can also employ a learning method based on the Expectation-Maximization algorithm in order to stably obtain its solutions [14].

4 Expected Influence Curve Acquisition

Thus far, we assumed that the time-delay and diffusion parameters can vary with respect to nodes and links. However, as mentioned earlier, we address the problem of estimating the influence curves from single observed diffusion sequences. Thus, in order to avoid overfitting to the observed data, we place a constraint that the parameters are uniform on nodes and links throughout the network G. Therefore, we set $r_{u,v} = r$ and $\kappa_{u,v} = \kappa$ for any link $(u, v) \in E$ in case of the AsIC model and $r_v = r$ and $\omega_{u,v} = \kappa |B(v)|^{-1}$ for any node $v \in V$ and link $(u, v) \in E$ in case of the AsLT model, where note that $0 < \kappa < 1$ and $\omega_{v,v} = 1 - \kappa$. Namely, since parameter κ of the AsLT model can be interpreted as a kind of diffusion probability, we employ the same symbol as used in the AsIC model. Without this constraint there is no way to learn the parameters since we only have one sequence of observation that covers only a small part of existing links.

We describe our method for acquiring an expected influence curve under the AsIC and AsLT model. Assume that we have observed the following single information diffusion sequence from the information source node v_0 at time t_0.

$$d = \{(v_0, t_0),\ (v_1, t_1),\ \cdots,\ (v_T, t_T)\}$$

First, by using the method described in Section 3.1 or 3.2, we can learn a pair of model parameters, κ and r, from the observed diffusion sequence d. Next, by using the model described in Section 2.2 or 2.4, we obtain the following K sets of simulated diffusion sequences

$$s_k = \{(v_0, t_0),\ (v_{k,1}, t_{k,1}),\ \cdots,\ (v_{k,T}, t_{k,T})\},\ k = 1,\ \cdots,\ K.$$

Note that the information source node v_0 at time t_0 is the same for all sequences, but their final activation times $\{t_{k,T}\}$ as well as their numbers of activated nodes $\{|s_k|\}$ vary in quite wide range, as shown later in our experiments. Finally, by using the generated sequences $S = \{s_1,\ \cdots,\ s_K\}$, we can estimate the expected influence curve $\sigma(t, v_0)$ as follows:

$$\sigma(t; v_0, d) = \frac{1}{K} \sum_{k=1}^{K} |\{(v, \tau) \in s_k\ ;\ \tau \leq t\}| \tag{9}$$

This method needs three kinds of input information, i.e., the single observed diffusion sequence d, the topology of observed social network G, and the number of diffusion simulation trials K; then it outputs the expected influence curve $\sigma(t, v_0)$. Below we summarize the estimation algorithm.

step 1. Learn a pair of parameters κ and r from d.
step 2. Generate $S = \{s_1, ..., s_K\}$ by simulating information diffusion K times with the learned parameters κ and r.
step 3. Calculate the expected influence curve $\sigma(t; v_0, d)$ as the average of S.

In our experiments, the number of diffusion simulation trials is set to $K = 100$.

5 Experiments

We evaluate the feasibility of the proposed estimation method using the topologies of two large real network data.

5.1 Evaluation Procedure

Below we describe a procedure to evaluate our proposed method.

proc. 1 Decide information diffusion model: AsIC or AsLT, and choose its true parameters κ^* and r^*, and an information source node v_0 at time t_0.
proc. 2 Generate a set of N diffusion sequences D under the setting of proc. 1.
proc. 3 Calculate the expected influence curve $\sigma(t; v_0)$ from D (by Equation (9) with S replaced by D) and the empirical influence curve $\varphi(t; v_0, d_n)$ from each $d_n \in D$.
proc. 4 Estimate the expected influence curve $\sigma(t; v_0, d_n)$ from each $d_n \in D$ by the proposed method in Section 4.
proc. 5 Calculate the RMSE curves E_C and E_D for evaluation.

In reality it is almost impossible to obtain the actual expected influence curve from observation. Thus our evaluation resorts to experiments based on synthetic data by assuming an information diffusion model, AsIC or AsLT, with a pair of model parameters, κ^* and r^* which we assume to be true (proc. 1). Then, by performing simulation based on the model with the true parameters, we can prepare a set of N synthetic diffusion sequences denoted by $D = \{d_1, \cdots, d_N\}$ (proc. 2). Next, by applying Equation (9) with respect to D (instead of S), we can obtain a reasonably accurate expected influence curve $\sigma(t; v_0)$ (proc. 3). Here we can also obtain an empirical influence curve for each of the generated sequence d_n defined by $\varphi(t; v_0, d_n) = |\{(v, \tau) \in d_n ; \tau \leq t\}|$ (proc. 3)[1]. On the other hand, by regarding each of the generated sequence d_n as a single observed diffusion sequence, we can estimate the expected influence curve $\sigma(t; v_0, d_n)$ by our method proposed in Section 4 (proc. 4). Finally, we evaluate the average accuracy of the expected influence curves estimated by our method by means of the RMSE (Root Mean Squered Error) curve $E_C(t)$ and compare it with that of the empirical influence

[1] Note that d_n is not continuous but $\varphi(t; v_0, d_n)$ is continuous with respect to t.

curves denoted by $E_D(t)$. Here these RMSE curves, $E_C(t)$ and $E_D(t)$, are defined as follows.

$$E_C(t) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\sigma(t; v_0, d_n) - \sigma(t; v_0))^2}, \quad E_D(t) = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\varphi(t; v_0, d_n) - \sigma(t; v_0))^2}.$$

We can consider that the RMSE curve for $E_D(t)$ corresponds to the average accuracy of the single observed diffusion sequence when we interprete it as a piece of crude knowledge.

5.2 Experimental Settings

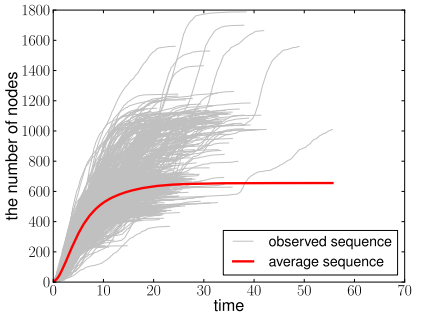
In our experiments, we employed two datasets of large real networks used in [8], which exhibit many of the key features of social networks. The first one is a trackback network of Japanese blogs. The network data were collected by tracing the trackbacks from one blog in the site *goo*[2] in May, 2005. We refer to this network data as the blog network. The blog network was a strongly-connected bidirectional network, where a link created by a trackback was regarded as a bidirectional link since blog authors establish mutual communications by putting trackbacks on each other's blogs. The blog network had 12,047 nodes and 79,920 directed links. The second one is a network of people that was derived from the "list of people" within Japanese Wikipedia. We refer to this network data as the Wikipedia network. The Wikipedia network was also a strongly-connected bidirectional network, and had 9,481 nodes and 245,044 directed links.

We determined the values of r and κ of the two models which we assumed to be true in the following way. In the AsIC model, we calculated the mean out-degree $\bar{d}$ and set two different values of κ in reference to $1/\bar{d}$, one smaller than $1/\bar{d}$ according to [7] and the other larger than $1/\bar{d}$ to see how a different value affects the result. Since the values of $\bar{d}$ were about 6.63 and 25.85 for the blog and the Wikipedia networks, respectively, the corresponding values of $1/\bar{d}$ were about 0.15 and 0.03. Thus, we decided to set $\kappa = 0.1$ and 0.3 for the blog network and $\kappa = 0.03$ and 0.09 for the Wikipedia network as the true values. As for the time-delay parameter r, we simply decided to set it to 1.0 because changing r is equivalent to changing the time scale accordingly. In the AsLT model, we only chose one value for κ. This is because we found that the information does not reach out far in the AsLT model and we needed to set a large value for κ to realize a decent diffusion. A value of 0.9 was a proper choice for κ. The time-delay parameter was set to $r = 1.0$, same as for the AsIC model.

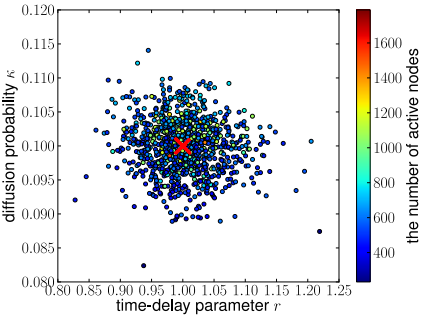
5.3 Experimental Results

blog network under the AsIC model. Figure 1 is the results of blog network under the AsIC model for the parameters $\kappa = 0.1$ and $r = 1.0$ (proc. 1). Figure 1(a) plots individual sequence data when the diffusion simulation was repeated $N = 1000$ times starting

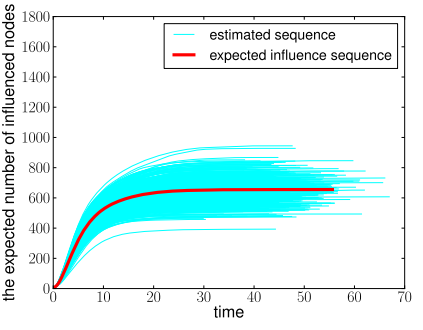
[2] http://blog.goo.ne.jp/



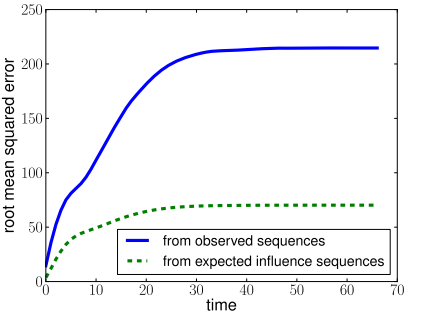
(a) Curves for generated sequences



(b) Parameter estimation results



(c) Curves from estimated parameters



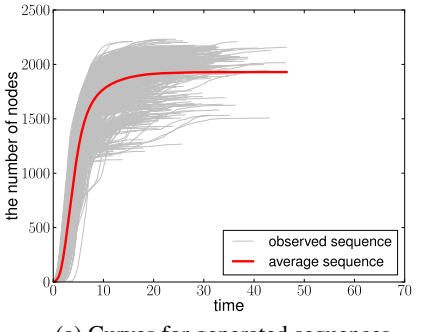
(d) Accuracy comparison

Fig. 1. The result set of blog network under the AsIC model ($\kappa^* = 0.1$)

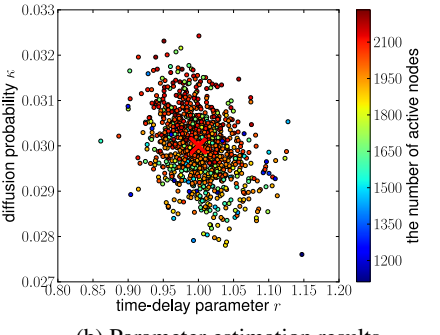
from the same initial source node (proc. 2). The horizontal axis is the time and the vertical axis is the number of active nodes. As shown in the figure, we observe a wide variety of influence curves with respect to time (dipicted in grey) due to the stochastic nature of the AsIC model. Here our task is to estimate the expected influence curve (depicted in red (black)), which is approximated by the empirical mean of the 1000 gray curves (proc. 3). Figure 1(b) is to show that it is possible to estimate the parameters of the AsIC model, i.e. time-delay parameter r and diffusion probability κ even from a single diffusion sequence. There are 1000 dots and each dot is the esitmated results (r, κ) from the corresponding sequence (proc. 4). We observe that the parameter estimation results are scatterd around the true values $(r^*, \kappa^*) = (1.0, 0.1)$, which were used to generate each sequence. The color (greyness) in the bar on the right indicates the lengh of the sequence, and the results are not very sensitive to the length unless it is very short. Figure 1(c) shows the estimated influence curves (depicted in cyan (grey)), each of which is obtained by performing simulation $K = 100$ times from the correspondng initial source node using the AsIC model with the same parameters learned from the corresponding original diffusion sequence. The target expected influence curve is the same as in Figure 1(a). Figure 1(d) shows the RMSE (Root Mean Squared Error) curves for both the original influence $\varphi(t; v_0, d_n)$ (Figure 1(a)) and the estimated influence $\sigma(t; v_0, d_n)$

(Figure 1(c)) with respect to the target influence (proc. 5). As shown, we observe that the RMSE for the estimated curve is much smaller (less than 1/3) than the one for the original one. Thus, we can say that the estimated influence curve is much closer to the expected influence curve than the original curve. Similar result is obtained for the case of $\kappa^* = 0.3$.

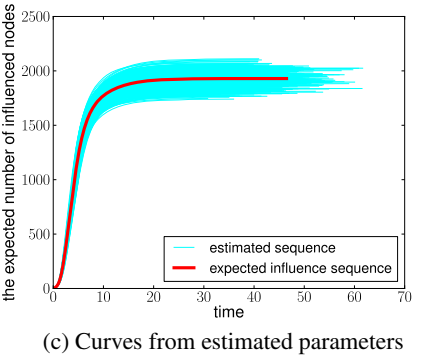
Wikipedia network under the AsIC model. Figures 2 and 3 are the results of Wikipedia network under the AsIC model for $\kappa^* = 0.03$ and $\kappa^* = 0.09$, respectively. In both cases, the RMSE for the estimated curve is much smaller (about 1/4 for $\kappa^* = 0.03$ and about 1/2 for $\kappa^* = 0.09$) in the proposed method. The results for $\kappa^* = 0.03$ is similar to the results of blog network except that the shape of the RMSE curve is different. However, the results for $\kappa^* = 0.09$ reveal different behaviors. When the diffusion probability is large, the information propagates far enough and individual sequence becomes similar to each other. Note that the number of nodes is almost doubled. The accuracy becomes better accordingly, especially for the original influence $\varphi(t; v_0, d_n)$. In general the proposed method is more effective when the diffusion probability is small and the observation sequences are diversified.



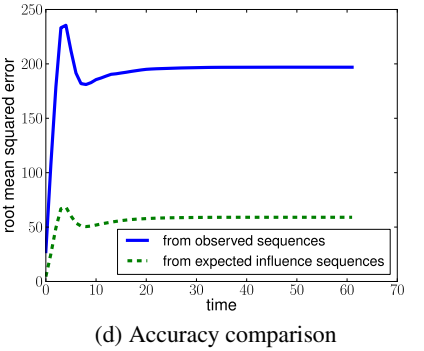
(a) Curves for generated sequences



(b) Parameter estimation results

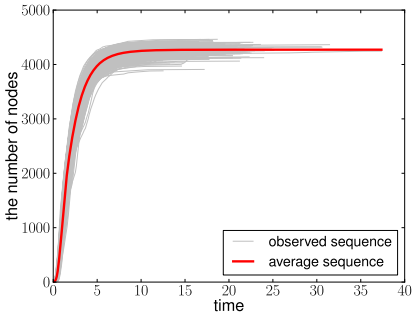


(c) Curves from estimated parameters

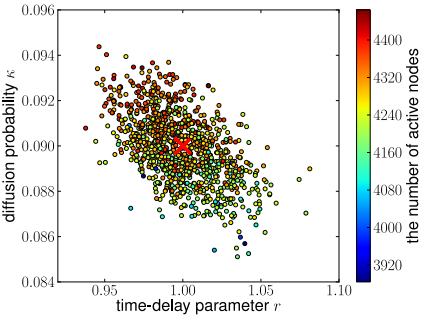


(d) Accuracy comparison

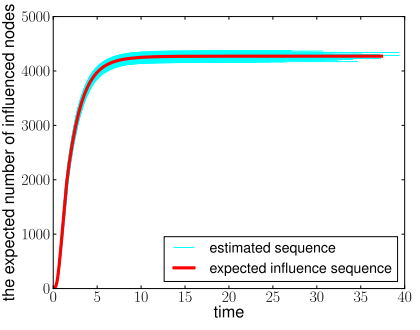
Fig. 2. The result set of Wikipedia network under the AsIC model ($\kappa^* = 0.03$)



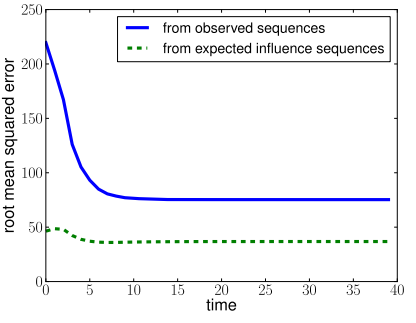
(a) Curves for generated sequences



(b) Parameter estimation results



(c) Curves from estimated parameters



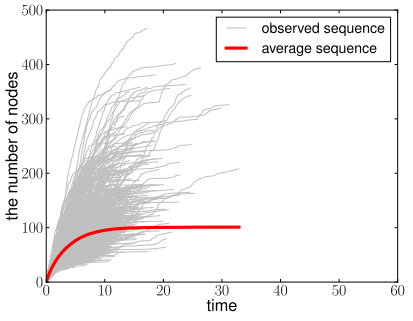
(d) Accuracy comparison

Fig. 3. The result set of Wikipedia network under the AsIC model ($\kappa^* = 0.09$)

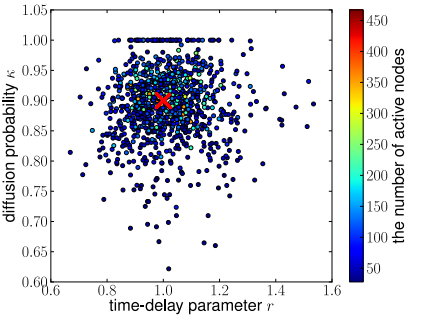
blog network under the AsLT model. Figure 4 shows the results of blog network under the AsLT model for $\kappa^* = 0.9$. Unlike the AsIC model, the information does not spread far and wide and the sequences are short. Accordingly the number of active nodes are much smaller (less than 500) and the errors in the parameter estimation are larger than the AsIC model. But still, we can say that the parameters are estimated reasonably well and the RMSE is much smaller (about 1/3) in the proposed method. Similar results are obtained for Wikipedia network.

5.4 Visual Analyses

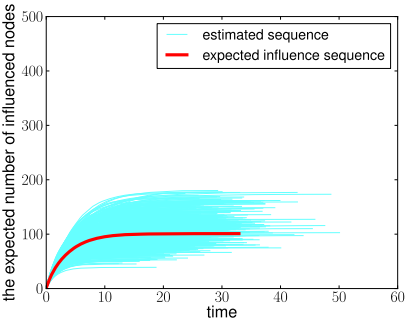
We saw that observation sequences are diverse in general due to the stochastic nature of the diffusion process. The differences in diffusion patterns are best understood by visualizing the active nodes. Figure 5 visualizes two extreme diffusion patterns for blog network of Figure 2 by using Cross-entropy method [15]. The red dots indicate active nodes and the gray dots non-active nodes. Figure 5(a) is the pattern for the longest sequence and Figure 5(b) is the one for the shortest sequence. We observe that dots are not uniformly distributed but have some dense reagions forming communities. In Figure 5(a) the information diffuses across many communities and spread widely, whereas



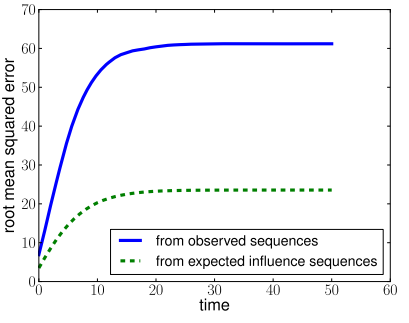
(a) Curves for generated sequences



(b) Parameter estimation results

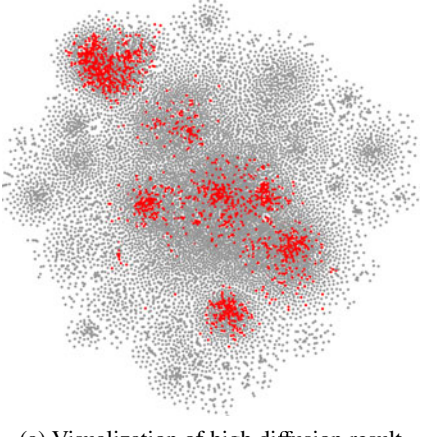


(c) Curves from estimated parameters

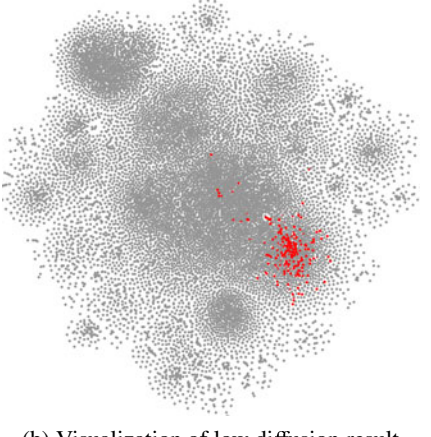


(d) Accuracy comparison

Fig. 4. The result set of blog network under the AsLT model ($\kappa^* = 0.9$)



(a) Visualization of high diffusion result



(b) Visualization of low diffusion result

Fig. 5. Visualization of blog network

in Figure 5(b) it is trapped within the same community of the initial source node and does not spread. Consequently, the number of active nodes in Figure 5(a) is 1,789 and that in Figure 5(b) is only 220. Simialr result is also observed in Wikipedia network.

6 Discussion

We note that the analysis we showed in this paper is the simplest case where κ and r take a single value each for all the links in E. However, the method is very general. In a more realistic setting we can divide E into subsets $E_1, E_2, ..., E_N$ and assign a different value κ_n and r_n for all the links in each E_n. For example, we may divide the nodes into two groups: those that strongly influence others and those not, or we may divide the nodes into another two groups: those that are easily influenced by others and those not. We can further divide the nodes into multiple groups. In this setting we learn κ_n and r_n for $n = 1, 2, ..., N$ from a single observation sequence.

We aimed to estimate the expected influence curve assuming two different information diffusion models in this paper but the framework of the proposed method can be applied to other models as well as other measures. For example, if we are interested in how different opinions spread [16], we can use the Voter model and estimate the expected opiniton share curve under this framework. Which measure and model to use depends on the problem we want to solve and the evaluation must be based on a task-specific performance measure. We believe that results in this paper based on fundamental models are applicable to more realistic information diffusion model.

7 Conclusion

One of the challenges of social network analysis is to estimate the expected influence degree with respect to time (expected influence curve). Because of the stochastic nature of information diffusion, a single observation sequence is not reliable to use as an approximation of this curve. We proposed a novel method to estimate the expected influence curve with good accuracy from a single observed information diffusion sequence assuming two types of information diffusion models: the asynchronous independent cascade (AsIC) model and the asynchronous linear threshold (AsLT). The method first learns the model parameters from a single observation sequence and next use the learned model to estimate the expected influence curve. We showed that parameter learning from a single sequence is feasible and practical, and the estimated influence curve is much more accurate than using the observed sequence as its approximation by extensive experiments using two real world networks.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

1. Newman, M.E.J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. Physical Review E 66, 035101 (2002)
2. Newman, M.E.J.: The structure and function of complex networks. SIAM Review 45, 167–256 (2003)
3. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. SIGKDD Explorations 6, 43–52 (2004)
4. Domingos, P.: Mining social networks for viral marketing. IEEE Intelligent Systems 20, 80–82 (2005)
5. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. In: Proceedings of the 7th ACM Conference on Electronic Commerce (EC 2006), pp. 228–237 (2006)
6. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters 12, 211–223 (2001)
7. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2003), pp. 137–146 (2003)
8. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data 3, 9:1–9:23 (2009)
9. Watts, D.J.: A simple model of global cascades on random networks. Proceedings of National Academy of Science, USA 99, 5766–5771 (2002)
10. Watts, D.J., Dodds, P.S.: Influence, networks, and public opinion formation. Journal of Consumer Research 34, 441–458 (2007)
11. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI 2007), pp. 1371–1376 (2007)
12. Kimura, M., Saito, K., Nakano, R., Motoda, H.: Finding influential nodes in a social network from information diffusion data. In: Proceedings of the International Workshop on Social Computing and Behavioral Modeling (SBP 2009), pp. 138–145 (2009)
13. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Learning continuous-time information diffusion model for social behavioral data analysis. In: Zhou, Z.-H., Washio, T. (eds.) ACML 2009. LNCS, vol. 5828, pp. 322–337. Springer, Heidelberg (2009)
14. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Behavioral analyses of information diffusion models by observed data of social network. In: Chai, S.-K., Salerno, J.J., Mabry, P.L. (eds.) Advances in Social Computing. LNCS, vol. 6007, pp. 149–158. Springer, Heidelberg (2010)
15. Yamada, T., Saito, K., Ueda, N.: Cross-entropy directed embedding of network data. In: Proceedings of the 20th International Conference on Machine Learning (ICML 2003), pp. 832–839 (2003)
16. Kimura, M., Saito, K., Motoda, H., Ohara, K.: Learning to predict opinion share in social networks. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence, AAAI 2010 (2010)

Automatic Speech-Based Classification of Gender, Age and Accent

Phuoc Nguyen*, Dat Tran, Xu Huang, and Dharmendra Sharma

Faculty of Information Sciences and Engineering
University of Canberra, ACT 2601, Australia
{Phuoc.Nguyen,Dat.Tran,Xu.Huang,
Dharmendra.Sharma}@canberra.edu.au

Abstract. This paper presents an automatic speech-based classification scheme to classify speaker characteristics. In the training phase, speech data are grouped into speaker groups according to speakers' gender, age and accent. Voice features are then extracted to feature vectors which are used to train speaker characteristic models with different techniques which are Vector Quantization, Gaussian Mixture Model and Support Vector Machine. Fusion of classification results from those groups is then performed to obtain final classification results for each characteristic. The Australian National Database of Spoken Language (ANDOSL) corpus was used for evaluation of gender, age and accent classification. Experiments showed high performance for the proposed classification scheme.

Keywords: Speaker characteristics, gender classification, age classification, accent classification.

1 Introduction

Humans are very good at recognizing people, besides visual cues, they also heavily rely on auditory cues. Pretty quickly we can assess from speech alone a person's gender, age, accent, educational or cultural background which are speaker characteristics [1]. Classifying speaker characteristics is an important task in Dialog Systems, Speech Synthesis, Forensics, Language Learning, Assessment Systems, and Speaker Recognition Systems. In Human-Computer Interaction applications, the interaction between users and computers taking place at the speech-driven user interface. For example, Spoken Dialogs Systems provide services in domains of finance, travel, scheduling, tutoring, or weather. The systems need to gather automatically information from the user in order to provide timely and relevant services. Most telephone-based services today use spoken dialog systems to either route calls to the appropriate agent or even handle the complete service by an automatic system. Another example of Human-Computer Interaction application is Computer-aided Learning and Assessment systems. The systems provide interactive recording and playback of user's input speech,

* Phuoc Nguyen is postgraduate research student at University of Canberra, Faculty of Information Sciences and Engineering, Australia.

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 288–299, 2010.

feedback regarding acoustic speech features, recognizing the input, and interpreting interaction to act as a conversation partner. Besides customizing to the native language of the language learner, learning systems may have to be tailored towards particular accents, for example the E-Language Learning System program between the U.S. Department of Education and the Chinese Ministry of Education [1]. In Human-Centered applications, the computers stay in the background attempting to anticipate and serve people's needs. One example is Smart Room Environments in which computers watch and interpret people's actions and interactions in order to support communication goals. Another example is Speech Translation system whose task is to recognize incoming speech from the source language and translate the text of the recognizer output into text of the target language, and then synthesize the translated text to audible speech in the target language. The system needs to generate appropriate synthesized output based on the speaker's gender, age and accent. Beyond that, speaker characteristics need to be assessed in order to adapt system components, particularly the speech recognition front-end to the specific voice characteristics of the speaker and the content of what was spoken. This adaptation process has been proven to dramatically improve the recognition accuracy, which usually carries over favorably to the performance of the overall system. Recent systems rely on speaker adaptive training methods, which first determine the speaker's identity and then apply acoustic model adaptation based on the assumed identity. Some applications rely on broader speaker classes such as gender or age group to load pre-trained models [1].

Some investigations on speaker characteristics have been found in the literature. In [2] acoustic features were extracted and trained Gaussian mixture models to identify subjective elderly speakers. In [3] general acoustic and prosodic features were also used to train hidden Markov models to classify speaker's gender, age, dialect, and emotion. Experiments in [4] used four classifiers for separate recognition of age and gender. In [5]-[9], feature analysis was investigated, results showed prosodic features gain better performance over acoustic features while do not require linguistic features. For accent classification, we particularly focus on Australian accent. Although the accent is only spoken by a minority of the population, it has a great deal of cultural credibility. It is disproportionately used in advertisements and by newsreaders.

According to linguists, three main varieties of spoken English in Australia are Broad (spoken by 34% of the population), General (55%) and Cultivated (11%) [10]. They are part of a continuum, reflecting variations in accent. Although some men use the pronunciation, the majority of Australians that speak with the accent are women.

Broad Australian English is usually spoken by men, probably because this accent is associated with Australian masculinity. It is used to identify Australian characters in non-Australian media programs and is familiar to English speakers. The majority of Australians speak with the General Australian accent. Cultivated Australian English has some similarities to British Received Pronunciation, and is often mistaken for it. In the past, the cultivated accent had the kind of cultural credibility that the broad accent has today. For example, until 30 years ago newsreaders on the government funded ABC had to speak with the cultivated accent [11].

Current research on Australian accent and dialect is focusing on linguistic approach to dialect of phonetic study [12][13], classification of native and non-native Australian [14], or to improve Australian automatic speech recognition performance [15]. However, there is no research on automatic speaker classification based on the

three Australian accents of Broad, General, and Cultivated. There has not been a classification system that can classify persons based on their gender, age and accent simultaneously.

This paper presents a speech-based classification scheme to classify speaker characteristics. In the training phase, speech data are grouped into speaker groups according to speakers' gender, age and accent. There are 18 speaker groups which are combinations of 2 gender groups (female and male), 3 age groups (young, middle and elderly), and 3 accent groups (broad, general and cultivated). Speech processing was performed using open source openSMILE feature extraction [16]. There are 16 low-level descriptors chosen including zero-crossing-rate (ZCR) from the time signal, root mean square (RMS) frame energy, pitch frequency (normalised to 500 Hz), harmonics-to-noise ratio (HNR) by autocorrelation function, and mel-frequency cepstral coefficients (MFCC) 1-12 in full accordance to HTK-based computation [17]. Voice features are extracted as feature vectors and are used to train speaker group models with different techniques which are Vector Quantization (VQ), Gaussian Mixture Model (GMM), and *C*-Support Vector Classifiers (C-SVC). Fusion of classification results from those groups is then performed to obtain results for each gender, age and accent. The Australian National Database of Spoken Language (ANDOSL) corpus consisting of 108 speakers and 21600 long utterances was used for evaluation [18]. Experiments showed high performance for the proposed classification scheme.

The rest of the paper is organised as follows. VQ, fuzzy VQ, GMM and *C*-SVC classifiers are summarised in Section 2. Section 3 presents our experimental results and Section 4 concludes our work.

2 Classifiers for Gender, Age and Accent Classification

2.1 Vector Quantization

Vector quantisation (VQ) is a data reduction method, which is used to convert a feature vector set into a small set of distinct vectors using a clustering technique [19]. The distinct vectors are called code vectors and the set of code vectors that best represents the training vector set is called the codebook. The VQ codebook can be used as a speech or speaker model. Since there is only a finite number of code vectors, the process of choosing the best representation of a given feature vector is equivalent to quantising the vector and leads to a certain level of quantisation error. This error decreases as the size of the codebook increases, however the storage required for a large codebook is nontrivial. The key point of VQ modelling is to derive an optimal codebook which is commonly achieved by using the hard *C*-means (HCM) (k-means).

Let $X = \{x_1, x_2, ..., x_T\}$ be a set of T vectors, each of which is a d-dimensional feature vector extracted by digital speech signal processing. Let $U = [u_{it}]$ be a matrix whose elements are memberships of x_t in the ith cluster, $i=1,...,C$, $t=1,...,T$. Hard C-partition space for X is the set of matrices U such that

$$u_{it} \in \{0,1\} \quad \forall i,t, \qquad \sum_{i=1}^{C} u_{it} = 1 \; \forall t, \qquad 0 < \sum_{t=1}^{T} u_{it} < T \quad \forall i \tag{1}$$

where $u_{it} = u_i(x_t)$ is 1 or 0 according to whether x_t is or is not in the ith cluster, $\sum_{i=1}^{C} u_{it} = 1 \; \forall t$ means each x_t is in exactly one of the C clusters, and $0 < \sum_{t=1}^{T} u_{it} < T \quad \forall i$ means that no cluster is empty and no cluster is all of X because of $2 \le C < T$.

The HCM method is based on minimisation of the sum-of-squared-errors function as follows [19]

$$J_m(U, \lambda; X) = \sum_{i=1}^{C} \sum_{t=1}^{T} u_{it} d_{it}^2 \tag{2}$$

where $U = \{u_{it}\}$ is a hard C-partition of X, λ is a set of prototypes, in the simplest case, it is the set of cluster centers: $\lambda = \{\mu\}, \mu = \{\mu_i\}, i = 1, ..., C$ and d_{it} is the distance in the A norm (A is any positive definite matrix) from x_t to μ_i, known as a measure of dissimilarity

$$d_{it}^2 = \| x_t - \mu_i \|_A^2 = (x_t - \mu_i)' A (x_t - \mu_i) \tag{3}$$

Minimising the hard objective function $J_m(U, \lambda; X)$ in (2) gives

$$u_{it} = \begin{cases} 1 & d_{it} < d_{jt} \quad j = 1, ..., C, \; \forall j \ne i \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$\mu_i = \sum_{t=1}^{T} u_{it} x_t \Big/ \sum_{t=1}^{T} u_{it} \tag{5}$$

where ties are broken randomly.

2.2 Fuzzy Vector Quantization

Fuzzy Vector Quantization (FVQ) is a fuzzy partitioning of X into C fuzzy subsets or C clusters, $1 < C < T$. The most important requirement is to find a suitable measure of clusters, referred to as a fuzzy clustering criterion. Objective function methods allow the most precise formulation of the fuzzy clustering criterion. The most well known objective function for fuzzy clustering in X is the least-squares functional, that is, the infinite family of fuzzy C-means (FCM) functions, generalized from the classical within-groups sum of squared error function [20][21]

$$J_m(U, \lambda; X) = \sum_{i=1}^{C} \sum_{t=1}^{T} u_{it}^m d_{it}^2 \tag{6}$$

where $U = \{u_{it}\}$ is a fuzzy c-partition of X, each u_{it} represents the degree of vector x_t belonging to the ith cluster and is called the fuzzy membership function. For $1 \le i \le C$ and $1 \le t \le T$, we have

$$0 \le u_{it} \le 1 \quad \forall i, t, \qquad \sum_{i=1}^{C} u_{it} = 1 \; \forall t, \qquad 0 < \sum_{t=1}^{T} u_{it} < T \quad \forall i \tag{7}$$

$m \ge 1$ is a weighting exponent on each fuzzy membership u_{it} and is called the degree of fuzziness; other parameters are defined as seen in VQ.

Minimizing the fuzzy objective function J_m in (6) gives

$$u_{it} = \left[\sum_{k=1}^{c} (d_{it} / d_{kt})^{\frac{2}{m-1}} \right]^{-1} \tag{8}$$

$$\mu_i = \sum_{t=1}^{T} u_{it}^m x_t \Big/ \sum_{t=1}^{T} u_{it}^m \tag{9}$$

2.3 Gaussian Mixture Model

Since the distribution of feature vectors in X is unknown, it is approximately modelled by a mixture of Gaussian densities, which is a weighted sum of K component densities, given by the equation

$$p(x_t \mid \lambda) = \sum_{i=1}^{K} w_i N(x_t, \mu_i, \Sigma_i) \tag{10}$$

where λ denotes a prototype consisting of a set of model parameters $\lambda = \{w_i, \mu_i, \Sigma_i\}$, w_i, i = 1,..., K, are the mixture weights and $N(x_t, \mu_i, \Sigma_i)$, i = 1,..., K, are the d-variate Gaussian component densities with mean vectors μ_i and covariance matrices Σ_i

$$N(x_t, \mu_i, \Sigma_i) = \frac{\exp\left\{-\frac{1}{2}(x_t - \mu_i)' \Sigma_i^{-1} (x_t - \mu_i)\right\}}{(2\pi)^{d/2} \mid \Sigma_i \mid^{1/2}} \tag{11}$$

In training the GMM, these parameters are estimated such that in some sense, they best match the distribution of the training vectors. The most widely used training method is the maximum likelihood (ML) estimation. For a sequence of training vectors X, the likelihood of the GMM is

$$p(X \mid \lambda) = \prod_{t=1}^{T} p(x_t \mid \lambda) \tag{12}$$

The aim of ML estimation is to find a new parameter model $\overline{\lambda}$ such that $p(X \mid \overline{\lambda}) \geq p(X \mid \lambda)$. Since the expression in (12) is a nonlinear function of parameters in λ, its direct maximisation is not possible. However, parameters can be obtained iteratively using the expectation-maximisation (EM) algorithm [22]. An auxiliary function Q is used

$$Q(\lambda, \overline{\lambda}) = \sum_{i=1}^{T} p(i \mid x_t, \lambda) \log[\overline{w}_i N(x_t, \overline{\mu}_i, \overline{\Sigma}_i)] \tag{13}$$

where $p(i \mid x_t, \lambda)$ is the *a posteriori* probability for acoustic class i, i = 1,...,c and satisfies

$$p(i \mid x_t, \lambda) = \frac{w_i N(x_t, \mu_i, \Sigma_i)}{\sum_{k=1}^{c} w_k N(x_t, \mu_k, \Sigma_k)} \tag{14}$$

The basis of the EM algorithm is that if $Q(\lambda,\overline{\lambda}) \geq Q(\lambda,\lambda)$ then $p(X \mid \overline{\lambda}) \geq p(X \mid \lambda)$ [23]-[25]. The following re-estimation equations are found

$$\overline{w}_i = \frac{1}{T}\sum_{t=1}^{T} p(i \mid x_t, \lambda) \tag{15}$$

$$\overline{\mu}_i = \frac{\sum_{t=1}^{T} p(i \mid x_t, \lambda) x_t}{\sum_{t=1}^{T} p(i \mid x_t, \lambda)} \tag{16}$$

$$\overline{\Sigma}_i = \frac{\sum_{t=1}^{T} p(i \mid x_t, \lambda)(x_t - \overline{\mu}_i)(x_t - \overline{\mu}_i)'}{\sum_{t=1}^{T} p(i \mid x_t, \lambda)} \tag{17}$$

2.4 Support Vector Machine

The training data set is labeled as $\{x_i, y_i\}, i = 1,...,l$, $y_i \in \{-1,1\}$, $x_i \in R^d$. Support vector machine (SVM) using C-Support Vector Classification (C-SVC) algorithm will find the optimal hyperplane [26]:

$$f(x) = w^T \Phi(x) + b \tag{18}$$

to separate the training data by solving the following optimization problem:

$$\min \qquad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \tag{19}$$

subject to

$$y_i \left[w^T \Phi(x_i) + b \right] \geq 1 - \xi_i \text{ and } \xi_i \geq 0 \text{ , } i = 1,...,l \tag{20}$$

The optimization problem (19) will guarantee to maximize the hyperplane margin while minimize the cost of error. $\xi_i, i = 1,...,l$ are non-negative slack variables introduced to relax the constraints of separable data problem to the constraint (9) of non-separable data problem. For an error to occur the corresponding ξ_i must exceed unity (20), so $\sum_i \xi_i$ is an upper bound on the number of training errors. Hence an extra cost $C\sum_i \xi_i$ for errors is added to the objective function (19) where *C* is a parameter chosen by the user.

The Lagrangian formulation of the primal problem is:

$$L_P = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i - \sum_i \alpha_i \{ y_i (x_i^T w + b) - 1 + \xi_i \} - \sum_i \mu_i \xi_i \tag{21}$$

We will need the Karush-Kuhn-Tucker conditions for the primal problem to attain the dual problem:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \Phi(x_i)^T \Phi(x_i) \tag{22}$$

subject to:

$$0 \le \alpha_i \le C \qquad \text{and} \qquad \sum_i \alpha_i y_i = 0 \tag{23}$$

The solution is given by:

$$w = \sum_i^{N_S} \alpha_i y_i x_i \tag{24}$$

where N_S is the number of support vectors.

Notice that data only appear in the training problem (21) and (22) in the form of dot product $\Phi(x_i)^T \Phi(x_i)$ and can be replaced by any kernel K with $K(x_i, x_j) = \Phi(x_i)^T \Phi(x_j)$, Φ is a mapping to map the data to some other (possibly infinite dimensional) Euclidean space. One example is Radial Basis Function (RBF) kernel $K(x_i, x_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}$

In test phase an SVM is used by computing the sign of

$$f(x) = \sum_i^{N_S} \alpha_i y_i \Phi(s_i)^T \Phi(x) + b = \sum_i^{N_S} \alpha_i y_i K(s_i, x) + b \tag{25}$$

where the s_i are the support vectors.

The binary SVM classifiers can be combined to handle the multiclass case: One-against-all classification uses one binary SVM for each class to separate their members to other classes, while Pairwise classification uses one binary SVM for each pair of classes to separate members of one class from members of the other.

3 Experimental Results

3.1 ANDOSL Database

The Australian National Database of Spoken Language (ANDOSL) corpus [18] comprises carefully balanced material for Australian speakers, both Australian-born and overseas-born migrants. The aim was to represent as many significant speaker groups within the Australian population as possible. Current holdings are divided into those from native speakers of Australian English (born and fully educated in Australia) and those from non-native speakers of Australian English (first generation migrants having a non-English native language). A subset used for speaker verification experiments in this paper consists of 108 native speakers. There are 36 speakers of General Australian English, 36 speakers of Broad Australian English and 36 speakers of Cultivated Australian English in this subset. Each of the three groups comprises 6 speakers of each gender in each of three age ranges (18-30, 31-45 and 46+). So there are total of 18 groups of 6 speakers labeled *ijk*, where *i* denotes *f* (female) or *m* (male), *j* denotes *y* (young) or *m* (medium) or *e* (elder), and *k* denotes *g* (general) or *b* (broad) or *c* (cultivated). For example, the group *fyg* contains 6 female young general Australian English

speakers. Each speaker contributed in a single session, 200 phonetically rich sentences. All waveforms were sampled at 20 kHz and 16 bits per sample.

3.2 Speech Processing

In speaker characteristics feature research, prosodic approaches attempt to capture speaker-specific variation in intonation, timing, and loudness 1. Because such features are supra-segmental (are not properties of single speech segments but extend over syllables and longer regions), they can provide complementary information to systems based on frame-level or phonetic features. One of the most studied features is speech fundamental frequency (or as perceived, pitch), which reflects vocal fold vibration rate and is affected by various physical properties of the speaker's vocal folds, including their size, mass, and stiffness. Distributions of frame-level pitch values have been used in a number of studies. Although they convey useful information about a speaker's distribution of pitch values, such statistics do not capture dynamic information about pitch contours and are thus not viewed as high-level here [1]-[9].

Speech processing was performed using open source openSMILE feature extraction 16. There are 16 low-level descriptors chosen including ZCR, RMS energy, pitch frequency, HNR, and MFCC 1-12 in full accordance to HTK-based computation. To each of these, the delta coefficients are additionally computed. Next the 12 functionals including mean, standard deviation, kurtosis, skewness, minimum and maximum value, relative position, and range as well as two linear regression coefficients with their mean square error (MSE) are applied on a chunk basis. Thus, the total feature vector per chunk contains 16 * 2 * 12 = 384 attributes.

3.3 Parameter Settings for VQ and FVQ

Because the feature values have different ranges and the Euclidean distance was used, the following normalization of features was applied:

$$x'_{tj} = \frac{x_{tj} - \mu_j}{s_j} \tag{26}$$

where x_{tj} is the j-th feature of the t-th vector, μ_j the mean value of all T vectors for feature j, and s_j the absolute standard deviation, that is

$$s_j = \frac{1}{T}\sum_{t=1}^{T} | x_{tj} - \mu_j | \tag{27}$$

In order to find good selection of number of clusters and to watch accuracy trend, various number of them are tried to conduct experiments. Result in Figure 1 shows that the highest accent classification rate is found when the number of clusters is 32.

3.4 Parameter Settings for GMM

GMM is regarded as one state continuous hidden Markov model, therefore we used HTK toolkit [17] to train and test GMMs. All feature vectors were converted to HTK format. The number of Gaussians was set to 32, which is equal to the number of code vectors in FVQ and VQ.

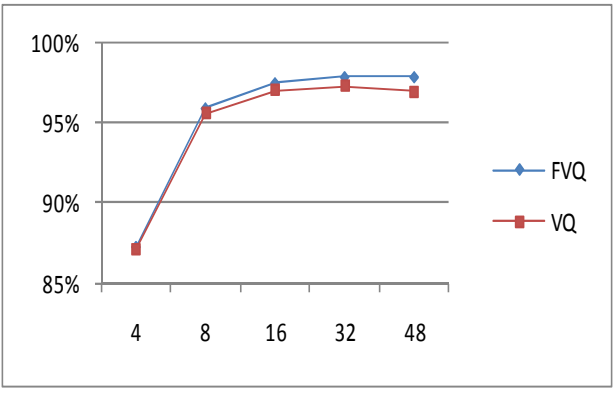


Fig. 1. Classification rates for FVQ (using Fuzzy *C*-Means) and VQ (using *K*-means)

3.5 Parameter Settings for SVM

Experiments were performed using WEKA data mining tool [27][28], C-SVC with RBF kernel were selected. All feature vectors were scale to range [-1, 1] in order to avoid domination of some dimension to general performance of classifiers. We performed several experiments with different values of parameters C and γ to search for the best model. The chosen values were $C = 2^1, 2^3, ..., 2^{15}$ and $\gamma = 2^{-15}, 2^{-13}, ..., 2^3$. The 10-fold cross-validation was used with every pair of values of C and γ. Results are shown in Figure 2 and we can see that the best values are $C = 2^7$ and $\gamma = 2^{-5}$.

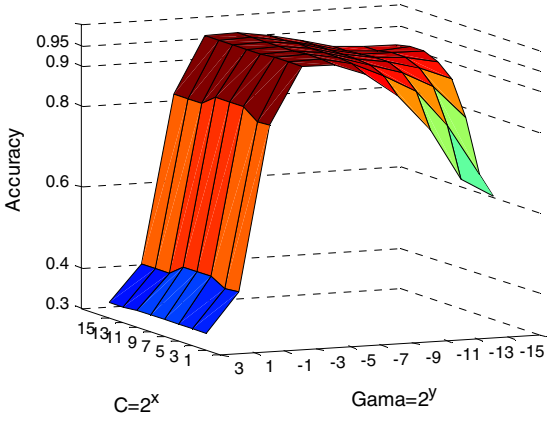


Fig. 2. Accent classification rates versus C and γ

3.6 Experimental Results

The result for all techniques and speaker characteristics are shown in Table 1.

Table 1. Speaker Classification Rates (%) for Gender, Age and Accent Classification

	Gender		Age			Accent		
	Male	Female	Young	Middle	Elderly	Broad	General	Cultivated
SVM	100.0	100.0	98.6	98.7	99.0	99.0	98.7	98.3
FVQ	100.0	99.9	98.6	98.1	98.7	98.7	98.4	98.2
VQ	99.9	99.9	97.9	97.9	98.0	98.2	98.1	97.6
GMM	100.0	99.9	96.7	96.7	97.6	97.2	96.7	96.7

Results showed that SVM achieved the best performance for all Gender, Age and Accent classifications. FVQ is better than VQ and GMM. As seen in the Introduction section, Cultivated Australian English has some similarities to British Received Pronunciation, and is often mistaken for it. The results in Table I also show that Cultivated classification achieved the lowest classification rate comparing with the other two accents Broad and General.

Tables 2, 3 and 4 present confusion matrices for each gender, age and accent classification. The total utterances are 21600. Table 2 shows very good result for gender classification. Table 3 shows reasonable errors. The number of utterances of young people misclassified as middle age people is higher than that misclassified as elderly people. Similar result is found for misclassified utterances of elderly people. Result for accent classification in Table 4 shows the lowest classification rate for Cultivated comparing with the other two accents Broad and General.

Table 2. Confusion Matrices for Gender Classification

	Male	*Female*
Male	10797	3
Female	1	10799

Table 3. Confusion Matrices for Age Classification

	Young	*Middle*	*Elderly*
Young	7097	53	50
Middle	50	7109	41
Elderly	32	37	7131

Table 4. Confusion Matrices for Accent Classification

	Broad	*General*	*Cultivated*
Broad	7128	27	45
General	25	7109	66
Cultivated	67	56	7077

4 Conclusion

We have presented gender, age and accent classification scheme using vector quantisation, Gaussian mixture modelling and support vector machine techniques. From the classification results for each speaker groups, a fusion technique was implemented to obtain final classification results for each of gender, age and accent characteristics. The proposed classification scheme was evaluated on the Australian speech database consisting of 108 speakers and 200 utterances for each speaker. Useful speech features were extracted. Most of classification rates were high and showed that the proposed classification scheme can be used in classification of speaker characteristics.

References

1. Schultz, T.: Speaker characteristics, in Speaker Classification I, pp. 47–74. Springer, Heidelberg (2007)
2. Minematsu, N., Sekiguchi, M., Hirose, K.: Automatic estimation of one's age with his/her speech based upon acoustic modeling techniques of speakers. In: Proc. IEEE Int'l Conference on Acoustic Signal and Speech Processing, pp. 137–140 (2002)
3. Shafran, I., Riley, M., Mohri, M.: Voice signatures. In: Proc. IEEE Automatic Speech Recognition and Understanding Workshop (2003)
4. Metze, F., Ajmera, J., Englert, R., Bub, U., Burkhardt, F., Stegmann, J., Müller, C., Huber, R., Andrassy, B., Bauer, J.G., Littel, B.: Comparison of Four Approaches to Age and Gender Recognition for Telephone Applications. In: ICASSP 2007 Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing, Honolulu, Hawai'i, USA, vol. 4, pp. 1089–1092 (2007)
5. Shriberg, E.: Higher-Level Features in Speaker Recognition, in Speaker Classification I, pp. 241–259. Springer, Heidelberg (2007)
6. Schötz, S.: Acoustic analysis of adult speaker age, in Speaker Classification I, pp. 88–107. Springer, Heidelberg (2007)
7. Campbell, J.P., Reynolds, D.A., Dunn, R.B.: Fusing high- and low-level features for speaker recognition. In: Proceedings of Eurospeech, pp. 2665–2668 (2003)
8. Schuller, B., Batliner, A., Seppi, D., Steidl, S., Vogt, T., Wagner, J., Devillers, L., Vidrascu, L., Amir, N., Kessous, L., Aharonson, V.: The relevance of feature type for the automatic classification of emotional user states: Low Level Descriptors and Functionals. In: Proc. Interspeech, Antwerp, pp. 2253–2256 (2007)
9. Schuller, B., Steidl, S., Batliner, A.: The INTERSPEECH 2009 Emotion Challenge. In: Proc. Interspeech. ISCA, Brighton (2009)
10. Mitchell, A.G., Delbridge, A.: The Pronunciation of English in Australia, pp. 11–19 (1965)
11. `http://www.convictcreations.com/research/languageidentity.html`
12. Harrington, J., Cox, F., Evans, Z.: An acoustic phonetic study of broad, general, and cultivated Australian English vowels. Australian Journal of Linguistics 17(2), 155–184 (1997)
13. Berkling, K., Zissman, M., Vonwiller, J., Cleirigh, C.: Improving accent identification through knowledge of English syllable structure. In: ICSLP 1998, pp. 89–92 (1998)
14. Kumpf, K., King, R.W.: Automatic accent classification of foreign accented Australian English speech. In: Fourth International Conference on Spoken Language Processing, pp. 1740–1743 (1996)

15. Kollengode, A.S., Ahmad, H., Adam, B., Serge, B.: Performance of speaker-independent speech recognisers for automatic recognition of Australian English. In: Proceedings of the 11th Australian International Conference on Speech Science & Technology, Auckland, pp. 494–499 (2006)
16. Eyben, F., Wollmer, M., Schuller, B.: Speech and Music Interpretation by Large-Space Extraction (2009), http://sourceforge.net/projects/openSMILE
17. Woodland, P.C., Gales, M.J.F., Pye, D., Young, S.J.: Broadcast news transcription using HTK. In: Proc. ICASSP 1997, Munich, pp. 719–722 (1997)
18. Millar, J.B., Vonwiller, J.P., Harrington, J.M., Dermody, P.J.: The Australian National Database of Spoken Language. In: Proc. Int. Conf. Acoust., Speech, Signal Processing (ICASSP 1994), vol. 1, pp. 97–100 (1994)
19. Duda, R.O., Hart, P.E.: Pattern classification and scene analysis. John Wiley & Sons, Chichester (1973)
20. Tran, D., Ma, W., Sharma, D., Nguyen, T.: Fuzzy Vector Quantization for Network Intrusion Detection. In: IEEE International Conference on Granular Computing, Silicon Valley, USA, November 2-4 (2007)
21. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
22. Hathaway, R.: Another interpretation of the EM algorithm for mixture distribution. Journal of Statistics & Probability Letters 4, 53–56 (1986)
23. Huang, X.D., Lee, K., Hon, H., Hwang, M.: Improved acoustic modeling for the SPHINX speech recognition system. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Toronto, Canada, pp. 345–348 (1991)
24. Reynolds, D., Rose, R.: Robust text-independent speaker identification using Gaussian mixture speaker models. IEEE Trans. Speech Audio Processing 3(1), 72–83 (1995)
25. Wildermoth, B.R., Paliwal, K.K.: GMM based speaker recognition on readily available databases. In: Micro. Elec. Eng. Research Conf. 2003 (2003)
26. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Knowledge Discovery and Data Mining 2(2), 121–167 (1998)
27. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
28. Chang, C.-C., Lin, C.-J.: LibSVM: a library for sup-port vector machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm

MMG: A Learning Game Platform for Understanding and Predicting Human Recall Memory

Umer Fareed[1] and Byoung-Tak Zhang[1,2]

[1] School of Computer Science and Engineering
[2] Graduate Programs in Cognitive Science and Brain Science
Seoul National University, Gwanak-gu, Seoul 151-742, Korea
{ufareed,btzhang}@bi.snu.ac.kr

Abstract. How humans infer probable information from the limited observed data? How they are able to build on little knowledge about the context in hand? Is the human memory repeatedly constructing and reconstructing the events that are being recalled? These are a few questions that we are interested in answering with our multimodal memory game (MMG) platform that studies human memory and their behaviors while watching and remembering TV dramas for a better recall. Based on the preliminary results of human learning obtained from the MMG games, we attempt to show that the human memory recall improves steadily with the number of game sessions. As an example case, we provide a comparison for the text-to-text and text-image-to-text learning and demonstrate that the addition of image context is useful in improving the learning.

Keywords: human learning, memory recall, Bayesian inference, human cognition.

1 Introduction

With the significant advancement in the field of artificial intelligence and machine learning over the past two decades, people tend to consider machines to be the near future replacement for many human tasks [1, 2]. But still, one can think of many tasks that humans perform better than the machines. The human ability to make robust, flexible and reliable inferences from limited data is one of the indisputable cases where nobody is able to find any counter argument. Without any doubt, there is a vast gap between human and machine learning at many levels. Take the example of language acquisition; being one of the hardest tasks for the computing machines to decipher but humans succeed in learning language purely from linguistic input. What we need here is to develop and improve computational systems for bridging this gap between humans and machines. These developments surely can contribute towards clearer and deeper insight into human cognition principles.

For instance, when we consider the case wherein people recognize words so quickly and so accurately from noisy speech, we ought to investigate what helps them to parse a string of words from the underlying grammar. Most of the time, inadequate data severely limits the inferences which people make about the data. Many of these

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 300–309, 2010.

choices come from prior probabilities of word structures known to them in one context or another. This thought drives us to conclude that the nature of the constraints on human inferences that guide human learning could be formalized using probabilistic models of cognition [3, 4].

Recently, there has been much discussion about probabilistic models of cognition. These techniques include probabilistic graphical models, such as Bayesian networks [5], Markov random fields [6], and Markov logic networks [7]. More recent models include deep belief networks [8] and random hypergraph structures [9] for parallel associative memory.

In our proposal, we describe a multimodal memory game (MMG) platform that studies human learning and memory behaviors to play and the machine to learn in an interactive digital cinema environment. The game manager being the core component of the game is responsible for controlling the working and learning environment of the game. We try to focus on machine learning with the help of human players whom we give a few minutes TV program clip to watch. Later, we ask questions about the subsequent text for the given text or text-image pairs taken from the video corpus. The human player's answers to the text or text-image queries teach the machine in the learning process to generate the succeeding text for different learning scenarios. We plot the points scored by the human players against the number of sessions played by them and observe the learning accuracy (performance of accurate recall) for the human players. Our interest is in modeling this learning accuracy and investigating how this learning leads to a normalized behaviour for learned vision-language modality information.

The proposed framework of the multimodal game is flexible and can be adapted to a more complex situation where more modalities and additional users are incorporated. The game also provides a research platform to study the life-long learning process in a dynamic environment since the video data can be played in a varying scenarios. The use of multimodal memory game for the experiments is a step towards the realization that cognitive learning associated with the human memory can be better understood with practical implementation through machine learning processes [10, 11, 12].

2 Human Learning and the Bayesian Models

Human learning in the real world is inductive, i.e., the learner builds generalizations or makes predictions based on the limited information in hand. Moreover, latest computational models of visual perception and inductive inference have demonstrated that Bayesian framework can be the optimal choice for capturing human behaviors.

Bayesian models of human cognition have met with a lot of success in recent years. But the use of rational models in cognitive science has been limited to specific cognitive processes in perception, memory and language processing [13]. Most of these rational models have been derived using Baye's rule, a particular theorem of probability that gives the relation between one conditional probability and its inverse. In other words, it relates the probability of a hypothesis given observed evidence and the probability of that evidence given the hypothesis. For example, if for a specific hypothesis H, $P(H)$ is the prior probability of H that was inferred before new evidence, E became available, we can calculate the posterior probability of H given E as

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)} \tag{1}$$

where $P(E)$ is the priori probability of witnessing the new evidence E under all possible hypotheses and $P(E|H)$ is the conditional probability of seeing the evidence E if the hypothesis H happens to be true (likelihood).

In the next section, we will propose the use of Baye's rule for our learning model and how it estimates the assumptions of the learner in the form of priori, likelihood and hypothesis space.

3 Multimodal Memory Game (MMG)

3.1 MMG Architecture

The multimodal memory game (MMG) provides a learning platform for the human players as well as the machine learner. The game architecture (Fig. 1) incorporates a user interface (game manager) that enables the human players to interact with the game and helps the machine learner to learn from the player's behaviour. The game involves two modalities of vision and language. There are two human players, i.e. the text to text generator T2T and the text-image to text generator TI2T.

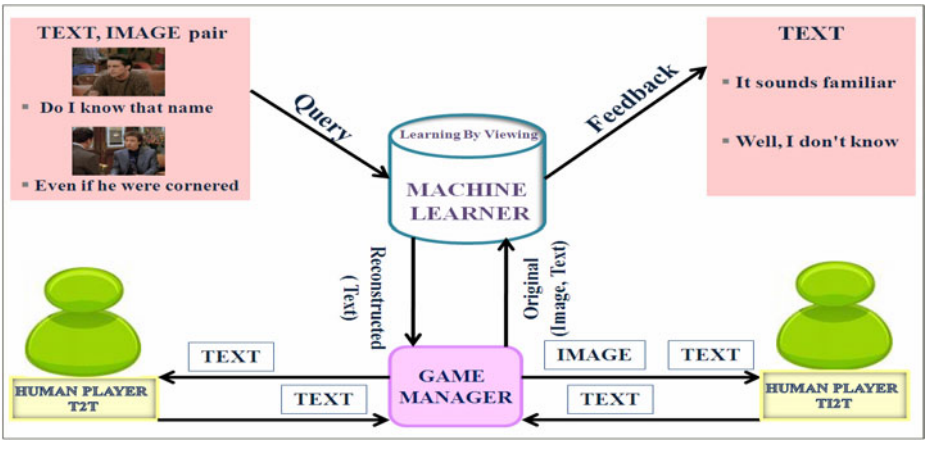


Fig. 1. Multimodal Memory Game Architecture

The game uses the multimodal information (image and text) from a video clip of a TV drama series, in this case, *Friends*. It proceeds in a quiz-like format. The game provide the human player with an interface for questions and answers, and the machine learner learns from the human input (and/or the correct answers which the game manager provide) while they play the game. We show the human players an episode of the selected TV drama after which they answer questions based either on a given text or text-image query taken from the video watched. We require the human player T2T to input the succeeding text for the queried text and similarly, the human player

TI2T provide the following text for the queried text-image pair. The game session continues in this manner until the user exits the game or answers all the questions required to attain a score that is recorded by the game manager.

3.2 MMG as a Cognitive Model of Language Learning

Recent research has provided ample evidence for use of Bayesian models of cognition to explain why humans behave as they do when presented with a specific task and the data to work with [14, 15, 16]. However, these models do not address human cognitive limitations in identifying what should be the optimal solution for the given data. They may use computational procedures that human learners cannot use.

In our proposed setup, the Bayesian learner seeks to learn from the language (incorporating the use of lexicon of words) input by the human player. Here the language acquisition is relatively easy for the Bayesian learner due to the fact that the human player has sound knowledge about the internalized structure of the observed data (English drama subtitles). We present the human player with some question $\boldsymbol{q}$, which is a segmented corpus of words and the player gives answer $\boldsymbol{a}$. This $(\boldsymbol{q}, \boldsymbol{a})$ builds the data $\boldsymbol{d}$ for machine learner. If we assume the human player to be the teacher and the machine learner to be the Bayesian learner in our framework, then the learner updates its hypothesis by the Baye's rule

$$P(h \mid d) \propto P(d \mid h)P(h) \tag{2}$$

where $P(h)$ is the learner's prior, $P(d|h)$ is the likelihood, and $P(h \mid d)$ is the posterior probability distribution of the hypothesis of the learner.

The Bayesian learner sees data produced by the teacher and forms a hypothesis about the approach used to produce that data. The learner then uses this hypothesis to produce data to make up its learning. For the Bayesian learner, the likelihood is 1 if the concatenated sequence of words in the hypothesis matches with those in the observed data. On the other hand, the likelihood is 0 if the sequence does not match the word sequence from the observed data.

In the above equation, we assume that the learner updates his belief with subsequent new examples and the teacher provides the learner with examples that tend to increase the learning. If the learner assumptions are in accordance with the observed data representation, the hypothesis is probable to have high prior probability. Varying these prior probabilities can result in characterization of new models which then can be used for the learners in a constrained learning environment.

4 The MMG Game Interface

We provide a user-friendly interface to the player that is easy to use and operate. The game offers the players to watch a video from the TV drama, for example a 20 minute-long episode of *Friends*. After watching the video clip, the player proceeds to choose one of the learning scenarios to play the game. For T2T learning, given text question, the player inputs the desired text. In TI2T learning, a text along with its visual context appears as a question and the player replies with the following text. Fig. 2 shows procedure for playing MMG games that gives an idea of how the game manager works.

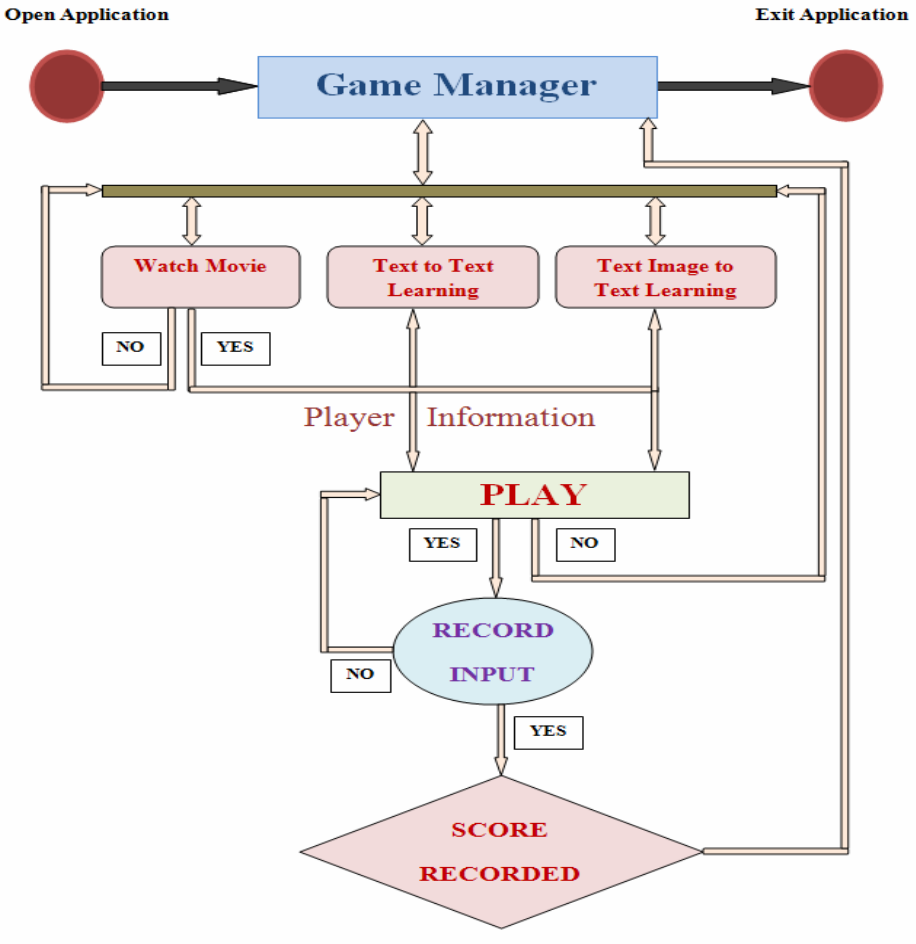


Fig. 2. Procedure for playing MMG games

When the human player clicks 'PLAY' to start the game, he provides personal information e.g. name, age and number of questions he would like to attempt during the game session. The players can choose the number of questions according to the need; they can be varied for multiple sessions or for longer sessions where the learning requires improvement. In case of text-to-text learning (T2T), a text from the TV drama episode appears on the screen and the player has to input the answer for queried text. The game interface for the text-to-text learning is shown in Fig. 3.

Similarly, in case of text-image-to-text (TI2T) query, an image along with text (subtitle) from the video clip appears on the screen. The player has to write the succeeding text to proceed to the next question as shown in Fig. 4. To make the learning process fast, the game time is managed and the players use a time quanta of 120 seconds for every question.

Fig. 3. User interface for text-to-text learning

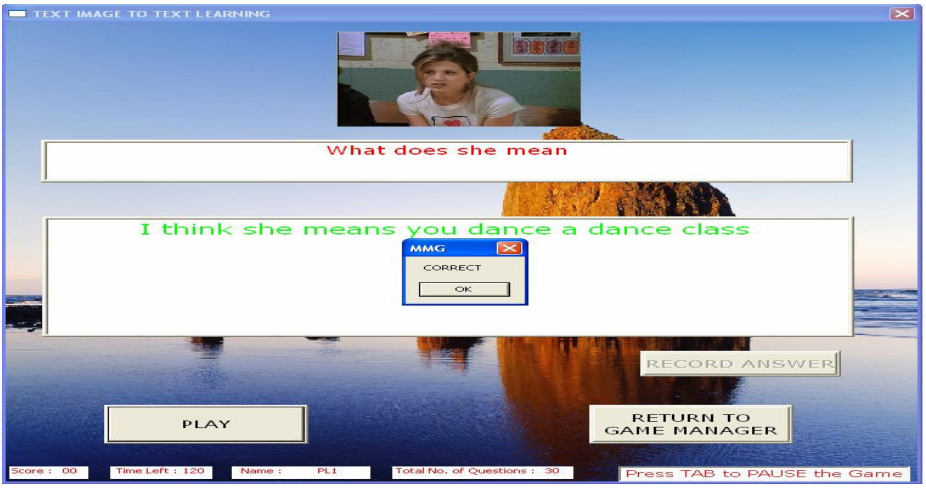


Fig. 4. User interface for text-image-to-text learning

During this time duration, if the player does not input the answer, the game manager proceeds to the next question. The game manager provides the user with the facility to pause the game by keyboard stroke in case he is interrupted by some external means. It updates the player status as the game proceeds along with the time left for the question on the status bar at the bottom of the interface. When the session ends, the game manager stores the player information with the number of correct answers in a text file for further use and reference.

5 Experimental Setup and Results

The data set for the experiments consists of the image-text pairs taken from an episode of the *Friends* series. We performed experiments for observing text-to-text learning and text-image-to-text learning obtained by the human players during the game play. The five volunteers were asked to watch the 23 minute video clip and play 10 sessions for each version of the game. The game manager scores sessions individually for all the players. We use the parameter settings for our experiments as given in Table 1.

Table 1. Parameters for the MMG experiments

Total no. of texts/images	No. of query texts/ images per session	No. of sessions	Time for each question (sec)
294	30	10	120

For scoring the experiments, we set a 60% threshold for the player input to be mark as correct based on the comparison of count of words that match correctly with the correct sequence of words.

Table 2. Comparison of player input to correct sentence

Player input	Correct sentence	Threshold (%)
I would say	I would have to say	60

Threshold in Table 2 is calculated using the below relation

$$\text{Threshold}\,(\%) = \frac{\text{Number of Words from Player Input}}{\text{Number of Words in Correct Sentence}} \tag{3}$$

The player input will be marked as correct if the ratio of the input words sequence to the correct word sequence matches the desired threshold. The result of each game session is used to calculate the accuracy of the human recall. We define recall accuracy as the ratio of the number of questions attempted correctly to the total number of questions in a given session as shown in below equation;

$$\text{Recall Accuracy} = \frac{\text{Number of Correctly Answered Questions}}{\text{Total Number of Questions}} \tag{4}$$

We use the average of the calculated accuracy for the five volunteers and plot the accuracy values against the number of sessions attempted. In the first experiment, we provide textual context as the cue and ask the players to respond with the succeeding text. In the second case of text-image-to-text learning, we add image context along with the

textual context of the video and asked the players to reply with the succeeding text for the queried image-text pair to observe the impact of image whether it aids in guessing the succeeding text. Similar parameter values are set for the text-image-to-text learning.

The volunteers play the game for 10 sessions and their scores are plotted against the number of sessions. The average accuracy curves for text-to-text learning and text-image-to-text learning in Fig. 5 shows a significant improvement in the human learning process from 5% to almost 60% during the 10 game sessions. The average accuracy curve for text-image-to-text learning produced a better response as compared to the text to text only case, however, the overall learning response for both the cases seem to have improved with the increasing number of game sessions.

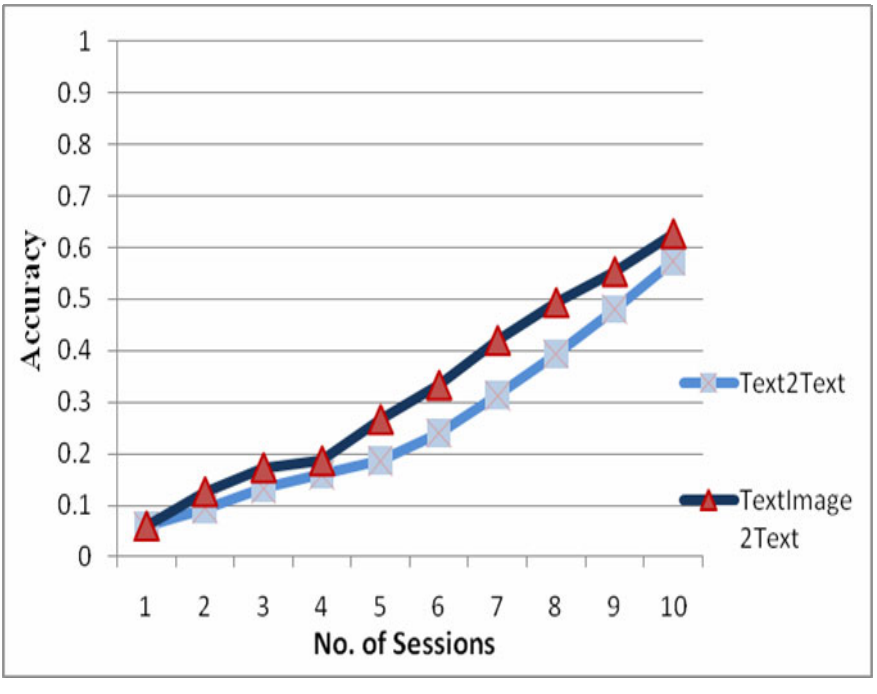


Fig. 5. Average of recall accuracy for text-to-text and text-image-to-text learning

Improvement in human learning can generally be achieved more rapidly by varying the game parameter settings, i.e. by exposing the human player to a larger portion (50 examples) of the training set (294 examples) in a single session rather than 30 questions may probably result in faster improvement (in term of the number of sessions) in learning response for the humans.

6 Concluding Remarks

We have presented an implementation of the multimodal memory game platform that can be used for studying the learning behaviors of humans and machines. Here we

proposed a simple cognitive model of learning that can be used for characterizing human learning. We anticipate the proposed model approach will be helpful in answering a few deep questions about human cognition; e.g. how human mind makes predictions from a limited observed data and what specific forms of knowledge do they acquire that support human inference in many different tasks.

From our experimental findings, we show that the human learning response improves gradually with the number of game sessions for both scenarios of text-to-text learning and text-image-to-text learning. We observed an apparent improvement in the learning curve for text-image-to-text learning which establishes our belief that addition of image is helpful in improving the learning. With these results, we believe it's too early to conclude about the general learning tendency of the humans, especially in our experimental setting where the player exposure is only to a small portion of the entire training set in each session.

We believe that this learning setup could form an inductive learning model of cognition and can be utilized to achieve human-like machine learning. Adoption of this procedure for machines may lead to life-long learning without complaints and it will be interesting to see how the human and machine learning performances compare for a larger scale experiments.

By setting the experimental parameters in a cognitively more plausible way, the model can be used for different learning platforms. These may include e-learning or robot learning in which the interaction is based on several different modalities of information in a changing environment.

Acknowledgment

This work was supported in part by IT R&D Program of MKE/KEIT (KI002138, MARS), NRF Grant of MEST (314-2008-1-D00377, Xtran), the BK21-IT program of MEST and the Higher Education Commission of Pakistan.

References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
2. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice Hall, Englewood Cliffs (2002)
3. Griffiths, T.L.: Connecting Human and Machine Learning via Probabilistic Models of Cognition. In: Technical Program. 10th Annual Conference of the International Speech Communication Association (2009)
4. Zhang, B.-T.: Cognitive learning and the multimodal memory game: Toward human-level machine learning. In: IEEE International Joint Conference on Neural Networks, pp. 3261–3267 (2008)
5. Jensen, F.V., Nielsen, T.: Bayesian Networks and Decision Graphs. Springer, New York (2007)
6. Vlontzos, J.A., Kung, S.Y.: Hidden Markov Models for Character Recognition. IEEE Trans. Image Processing 1(4), 539–543 (1992)
7. Richardson, M., Domingos, P.: Markov Logic Networks. Machine Learning 62, 107–136 (2006)

8. Sutskever, I., Hinton, G.E.: Deep Narrow Sigmoid Belief Networks are Universal Approximators. Neural Computation 20, 2629–2636 (2008)
9. Zhang, B.-T.: Hypernetworks: A Molecular Evolutionary Architecture for Cognitive Learning and Memory. IEEE Computational Intelligence Magazine 3(3), 49–63 (2008)
10. Maragos, P., Potamianous, A.: Multimodal Processing and Interaction Audio, Video, Text. Springer Science Media, Heidelberg (2008)
11. Benczúr, A., Bíró, I., Brendel, M.: Cross-Modal Retrieval by Text and Image Feature Biclustering. In: CLEF 2007: Proceedings of Cross Language Evaluation Forum (2007)
12. Zhang, R., Zhang, Z., Li, M., Ma, W.-Y., Zhang, H.-J.: A Probabilistic Semantic Model for Image Annotation and Multi-Modal Image Retrieval. Multimedia Systems 1, 27–33 (2006)
13. Oaksford, M., Chater, N.: Ten Years of the Rational Analysis of Cognition. Trends in Cognitive Science 3, 57–65 (1999)
14. Griffiths, T.L., Tenenbaum, J.B.: Structure and Strength in Causal Induction. Cognitive Psychology 51, 354–384 (2005)
15. Pearl, L., Goldwater, S., Steyvers, M.: How ideal are we? Incorporating human limitations into Bayesian models of word segmentation. In: Proceedings of the 34th Annual Boston University Conference on Child Language Development. Cascadilla Press, Somerville
16. Goldwater, S., Griffiths, T., Johnson, M.: Distributional Cues to Word Boundaries: Context is Important. In: BUCLD 31: Proceedings of the 31st Annual Boston University Conference on Language Development, pp. 239–250. Cascadilla Press, Somerville (2007)

Efficient Bulk-Insertion for Content-Based Video Indexing

Narissa Onkhum and Juggapong Natwichai

Computer Engineering Department
Faculty of Engineering
Chiang Mai University, Chiang Mai, Thailand
narissa.onkhum@gmail.com, juggapong@chiangmai.ac.th

Abstract. Videos have become one of the most important communication means these days. In this paper, we propose an approach to efficiently bulk-insert a set of new video index-entries into the existing video database for content-based video search. Given the current situation that enormous amount of new videos are created and uploaded to the video sharing websites, the efficient approaches are highly required. The environment we focused is where a B^+-tree is applied to index the video content-features. We propose a hybrid bulk-insertion approach based on a well-known bulk-insertion. Unlike the traditional bulk-insertion in which the traversals to insert the remaining index entries are performed to the ancestors, we propose to add a leaf-level traversal to improve the efficiency. Thus, our approach works in a hybrid manner, i.e., it switches between the leaf and ancestor traversals with regard to a condition with a very small additional cost. The experiments have been conducted to evaluate our proposed work by comparing to the one-by-one insertion approach, and the traditional bulk-insertion approach. The experiment results show that the proposed approach is highly efficient for video content-based indexing.

1 Introduction

Advances in video processing technologies have made videos become one of the most important communication means currently. These circumstances have been boosted by the increasing speed of the Internet. As a result, streaming videos services on the video sharing websites, e.g. Youtube or Google Video, are growing rapidly. Using such services, the users can not only watch the videos, but also can create and share their own videos. In [1], it has been reported that there are approximately 150,000 videos uploaded to YouTube each day. Obviously, searching and indexing on such enormous size of these video databases are not a trivial task.

Typically, the video database systems can be categorized by their search methodology into keyword search and content-based search. Although, the mentioned popular services are considered to be in the first category. There are a lot of applications in which the keyword search mechanism can not provide. For

B.-H. Kang and D. Richards (Eds.): PKAW 2010, LNAI 6232, pp. 310–321, 2010.

example, searching for the similar videos given an original video in copyright-law enforcement [2]. This application needs to consider the video contents, not the keywords. In content-based video search, the features from the content of each video in the database are first extracted and stored. When a query video is given for a search, the similarity between the query video and the videos in the database will be computed. Only the best matched videos under a pre-specified similarity-criteria are returned as the result.

Given that the size of the video database tends to be very large, the performance of the content-based video search might be inefficient, unless the appropriate approaches are applied. Video indexing is one of the most important approaches for efficiency improvement on these contexts. Typically, the summarization of each video in the database is first determined. Subsequently, an index is built using the summarization data as the keys. In [3,4], the authors proposed to use a well-known B^+-tree index for such task. The key of a B^+-tree represents a cluster of similar videos. Thus, given a query video, the key of it can be used to find the cluster of similar videos by such access path efficiently. According to the experiment results, search time of this approach can be reduced up to two times comparing with the traditional approach.

However, the search efficiency might not be the only one important problem for the content-based video search. According to the mentioned fact, the volume of the videos to be inserted to the video sharing websites can be extremely large, and it can be even larger in the future. Though, the existing video index approaches are efficient for the search operations. The efficiency of the insertion cannot be neglected. Furthermore, since the nature of the B^+-trees are to evolve itself for balancing when each index entry is inserted. The efficiency might be degraded in the current circumstance.

For example, consider the B^+-tree in Figure 1. Suppose that it can store up to four index entries at a leaf node, also the order of the tree is 4. If an index-entry with the key '1.97' is to be inserted into the tree as presented, first, the path from the root to the appropriate leaf will be determined. We can see that such determination requires three disk access. Subsequently, the leaf will be split since there is no room for the new entry, and such split will be propagated to the ancestors of the leaf. The overall disk access in order to balance the index

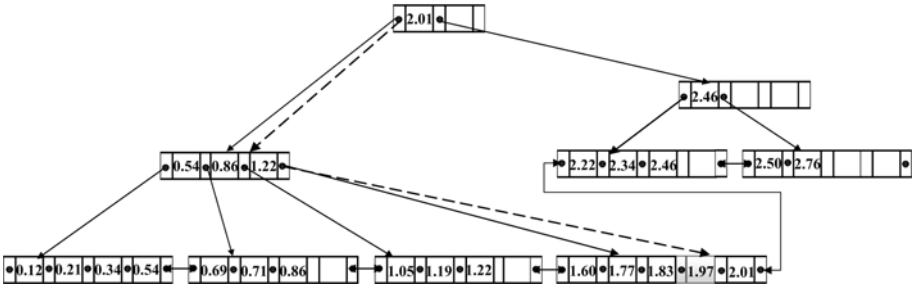


Fig. 1. An Example of the B^+-tree

when the only one index entry is inserted could be at least nine times. Obviously, the one-by-one insertion approach can be inefficient when the large amount of video index-entries are to be inserted. An alternative approach for this efficiency problem is batch-insertion [5], However, such approach requires the index-tree to be re-built in a bottom-up manner. This might not be desirable for the on-line application such as video sharing. Furthermore, its efficiency might also be questionable.

In this paper, we address the problem of video-index insertion on the efficiency aspect. Instead of one-by-one index insertion, our proposed approach is based on bulk insertion. We apply a well-known bulk-insertion approach proposed in [6]. The approach first sorts index entries to be inserted using their keys in the main memory. Subsequently, the partitioned sorted-keys indexes are formed in which each partition contains only the keys to be inserted at the same index block. Before the insertions, every partition will be transformed into a subtree. Finally, the subtrees are merged with the main B^+-tree. Since the keys of a subtree can be inserted at once into the same index block, the number of the disk accesses can be reduced enormously. Furthermore, we also proposed an improvement for the bulk-insertion approach to suit the video indexing for the content-based video search. Typically, if all the index entries in a bulk cannot be inserted at the index block entirely, the traditional bulk-insertion approach will continue inserting the remaining partitions by traversing up to the ancestors of the current node to find the internal node which can cover the remaining keys. If the gaps between the keys in a bulk are not large, this ancestor-level traversal might not be efficient. Therefore, we propose another mode of index-traversal, i.e., leaf-level traversal to improve the efficiency. The improved bulk-insertion approach works in a hybrid manner, i.e., it switches between the leaf and ancestor traversals with regard to a condition with a very small additional cost. The experiments have been conducted to evaluate our proposed work by comparing to the one-by-one insertion approach, and the traditional bulk-insertion approach.

The main contributions of this paper are summarized below.

- The bulk-insertion in [6] is realized in the context of video content-based search.
- An improvement for the bulk-insertion approach for the video content-based search is proposed.
- We also present the experiment results on the video databases with various characteristics. The results demonstrate that our proposed work is more efficient than the bulk-insertion without the improvement, as well as the one-by-one insertion approach.

The rest of this paper is organized as follows. Section 2 briefly reviews content-based video indexing. In Section 3, our proposed approach to bulk-insert the video index is presented. The experiment results of the approach are reported in Section 4. Section 5 reviews the related work. Finally, Section 6 gives the conclusion and discusses the future work.

2 Content-Based Video Indexing

Typically, the approaches to improve the content-based video search may be categorized into two major groups. First, instead of processing videos in the traditional key-frames basis [7], there are many attempts have been proposed to summarize the content/feature of the videos for reducing the computational expense [3,8]. Subsequently, similarity measurement between two videos in the summarization form can be efficiently computed. Secondly, the indexing is applied to provide the auxiliary access path to the video content.

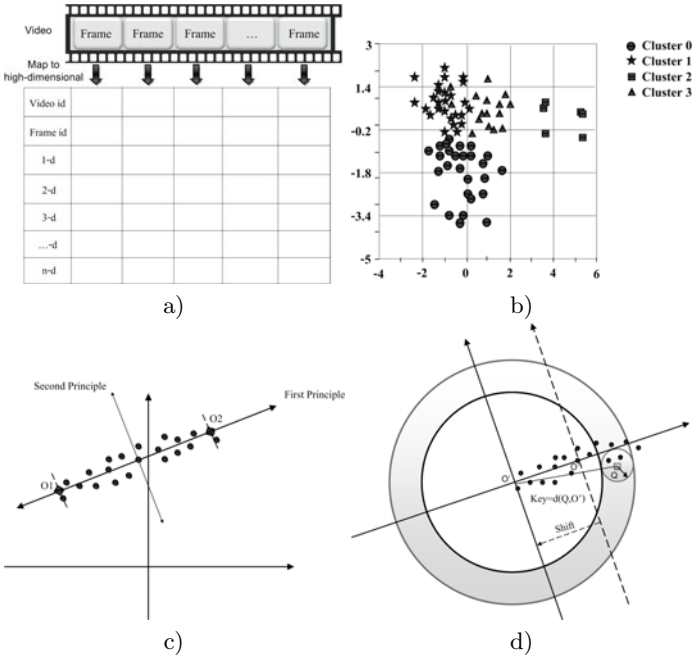


Fig. 2. Video Indexing Process

In this section, we present a prominent video indexing for content-based search, which was initially proposed in [3]. Based on such work, we will subsequently proposed our work to improve the efficiency when the index entries are bulk-inserted in the next section.

Such video indexing approach starts with transforming each video into a high-dimensional data tuple as in Figure 2a). Each dimension is a selected feature of the video databases, e.g., a range of histogram from the color space. After the transformation, the clusters of similar frames of the videos are determined as shown in Figure 2b). The description of each cluster, called Vitri, is considered as a compact representation of its. When giving two Vitris, the similarity between them can be determined by the estimated number of similar frames shared between the two clusters. Thus, the similarity of the two videos is determined

using the estimating total number of similar frames. Subsequently, as shown in Figure 2c), the high-dimensional features are mapped into a single dimension feature such that the inter-distance of the clusters is preserved as much as possible. Last, a B^+-tree is built using the single dimension feature of the cluster as a key, and provides the efficient access path to the video search operation as in Figure 2d).

3 Video Index Bulk-Insertion

It can be seen from the previous section that B^+-tree index needs to be evolved when a new video is inserted. In this section, the applying of the selected approach for index bulk-insertion for content-based video search is presented. The basic approach which was first proposed in [6] will be presented in Section 3.1. Eventually, our efficiency improvement tailored for the video search is proposed in Section 3.2.

3.1 Basic Approach

The approach is composed by two main steps, i.e., insertion step following by rebalancing step. The algorithm of the first step for inserting a bulk B with m keys into the main B^+-tree index, T, is shown in Figure 3. It begins with searching for the insertion location. Subsequently, it builds and inserts the sub-B^+-tree. After each insertion, it traverses back to the root until the node, which can cover the next insertion is found. Then, the insertion is repeated until the keys in the bulk are all inserted.

After the set of subtrees are constructed and merged with the main B^+-tree, the next step is to rebalance the tree (from the Line 14 of the algorithm in Figure 3). The algorithm for rebalancing the index after the subtree B_i of a bulk is inserted is shown in Figure 4. It begins with splitting the root of B_i into the left-part p_l, and the right-part p_r. Then, the merging nodes on the left, q_l, and the right, q_r, are determined, and merged with p_l and p_r respectively. Such rebalancing process is continued for the upper levels.

3.2 Efficiency Improvement

From the bulk-insertion presented in the previous section, it can be seen that the entries insertion of a bulk may need to traverse up to the ancestors of the current insertion node (from Line 11-12 of Figure 3). It might be the case that the traversal can continue up to the root of the index-tree.

In video content-based search context, we have found in our preliminary experiments that the bulks of index entries with moderate size built to be inserted typically covers only small part of the existing keys of the video databases. For example, a bulk with 50 videos may cover only 5-10 keys in the index of the videos database with 500 videos having 80 keys. It means that the gaps between the keys in typical bulks can be small.

```
1 Let p be the root of T.
2 Let S be an empty stack.
3 Let i=0.
4 while (i ≤ m)
5   Search for the insertion place l_i of key B.k_i.
6   Push such nodes into S.
7   Construct a B+-tree from the keys k_i, ..., k_{i+j},
      k_{i+j} ≤ l_i.highestkey, k_{i+j} ≥ k_other, for all k_other ∈ B.
8   Replace l_i with B_i.
9   i = i + j + 1.
10  If i ≤ m then
11    Pop nodes from S until the popped node pn covers the new pivot key k_i.
12    Set pn as the new root.
13  else
14    Rebalance all B_i
15    break;
16  end if
17 end while
```

Fig. 3. Bulk-insertion Algorithm

```
1 Let h = 1.
2 Let the parent of the root of B_i be denoted as p.
3 do
4   Split p into p_l and p_r, such that p_l contains pointers to the children
      storing keys smaller than the smallest key in B_i and p_r contains
      pointers to the children storing keys larger than the largest key in B_i.
5   Let p = p_l.
6   Merge p_l with q_l.
7   Merge p_r with q_r.
8   Adjust the pointers in the height h according to the loading factor
      of the B+-tree appropriately.
9   Let h = h + 1.
10  Let p be the node that the lowest key in its subtree is smaller than
      the smallest key in B_i, and the highest key in its subtree is larger
      than the largest key in B_i.
11 while h < B_i.height
12 Link the root of B_i to the corresponding pointer.
```

Fig. 4. Rebalancing Algorithm

Therefore, we propose another mode of index-traversal, i.e., leaf-level traversal to improve the efficiency. Such mode of traversal can work together with the traditional ancestor traversal in hybrid basis. In order to introduce the hybrid approach, for each leaf-level node, the pointer to the highest key of the next leaf node is stored. The example of a new index structure modified from Figure 1 is shown in Figure 6 as the highest key parts are hi-lighted. The hybrid algorithm

```
1 Let p be the root of T.
2 Let S be an empty stack.
3 Let i=0.
4 while (i ≤ m)
5   Search for the insertion place l_i of key B.k_i.
6   Push such nodes into S.
7   Construct a B+-tree from the keys k_i, ..., k_{i+j},
      k_{i+j} ≤ l_i.highestkey, k_{i+j} ≥ k_other, for all k_other ∈ B.
8   Replace l_i with B_i.
9   i = i + j + 1.
10  If i ≤ m then
11    If k_i ≤ l_i.next.highestkey then
12      Set l_i.next as the new root.
13    else
14      Pop nodes from S until the popped node pn covers the new pivot key k_i.
15      Set pn as the new root.
16    end if
17  else
18    Rebalance all B_i
19    break;
20  end if
21 end while
```

Fig. 5. Hybrid Bulk-insertion Algorithm

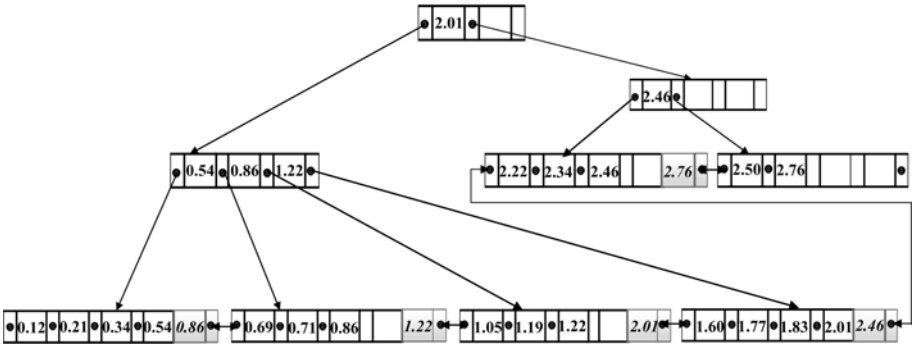


Fig. 6. A Hybrid Bulk-insertion Example

is presented in Figure 5. When the entries, which their keys are less than or equal to the highest key of the current leaf shown in Line 7 in Figure 5, are all inserted, the hybrid approach will decide which traversal mode to be selected. If the highest key of the next leaf node can cover the key of the next inserted entry, the leaf-level traversal will be selected as shown in Line 11 of Figure 5. From Figure 6, if the next inserted entry is 2.44 in which we can see that the next leaf node can cover the keys of the bulk, therefore, only one-hop traversal

is then proceeded. Otherwise, the traditional ancestor-traversal will be selected instead.

The efficiency of our hybrid approach is at least as efficient as the traditional approach, though the space complexity is increased by $\Theta(n)$ where n is the number of the leaves in the index. In the cases that the keys in a bulk to be inserted have small gaps, the leaf-level traversal mode will eliminate the time complexity of ancestor-traversal, which is at most the $O(h)$, where h is the height of the index-tree.

4 Experiment Results

In this section, we evaluate the performance of the proposed work. Our video database consists of 1000 real videos varying in length from 10 to 30 seconds. Such videos are processed as mentioned in Section 2 resulting in the 97 keys of the index entries. We evaluate the efficiency of our proposed work in term of the number of disk accesses required for the index insertion as in other index performance research literatures. The proposed work is compared with the bulk-insertion without the improvement, as well as the one-by-one insertion approach.

4.1 Size of Video Database Is Varied

In the first experiment, our proposed work will be evaluated in the scenarios which the number of the videos in the database is different. The results can present the efficiency when the size of database is scaled up. We set the size of bulk to be inserted at 10% of the size of databases, while the videos existing in the database are varied from 500 to 900 videos. In each experiment, the random

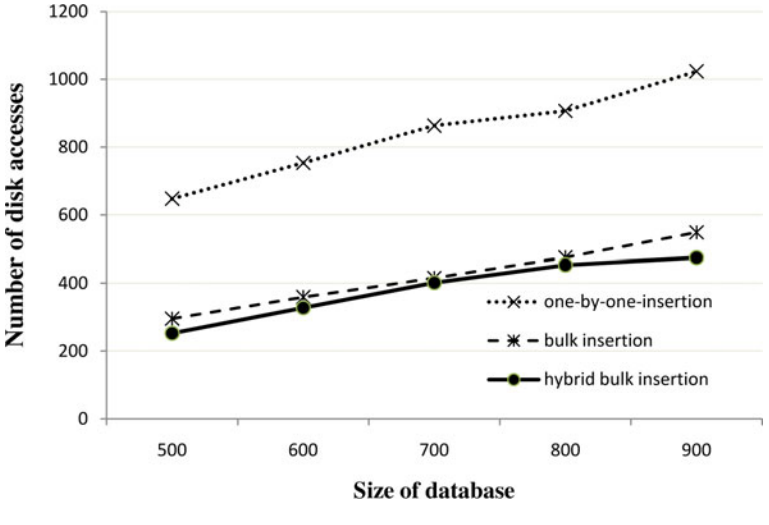


Fig. 7. Effect of the size of database on efficiency

set of the videos to be inserted is selected. The numbers reported in the result are five-time average.

Figure 7 shows the result of the experiment. We can see that the bulk-insertion algorithms both perform very efficiently comparing with the one-by-one insertion algorithm. Furthermore, we can see that the performance of the improved-hybrid algorithm is clearly better than or at least equivalent to the traditional bulk-insertion algorithm. The rationale behind this is the keys in the bulks to be inserted cover only small part of the existing indexes, thus the hybrid algorithm can benefit from the leaf-level traversal mode. Even when the size of the database is increased, the hybrid algorithm is still efficient since the size of the bulk is fixed at 10%, so the coverage of the keys to be inserted to the database can be considered as the same percentage.

4.2 Bulk Size Is Varied

In this experiment, we evaluate our work when the bulk size is increased. We separate 500 videos and use them as the fixed-size video database. Subsequently, a set of videos in the certain percentage of the database size is inserted in bulk. Note that the one-by-one insertion algorithm will insert each bulk in one-by-one manner. The bulks to be inserted are selected randomly in the experiments, and the five-time average efficiency numbers are reported.

The experiment result is shown in Figure 8. We can see that the proposed hybrid-bulk insertion algorithm is more efficient than the traditional algorithm. Overall, the disk accesses of the proposed algorithm are less than the traditional algorithm approximately 15 %. We also have found that the most different result between the proposed algorithm and the traditional algorithm is when the size of bulk to be inserted is set at 12 % of the size of the existing video database,

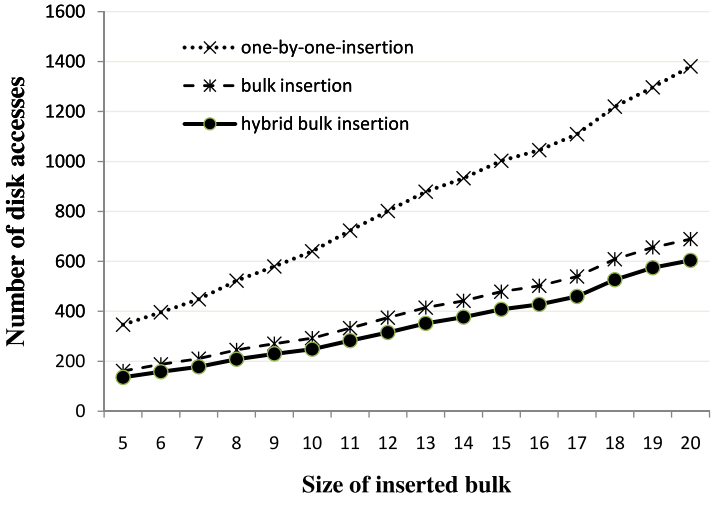


Fig. 8. Effect of the size of the inserted bulk on efficiency

i.e., the proposed algorithm uses less disk accesses by 20 % . Subsequently, the difference is started to decrease. This is because the keys of the bulk starts to cover larger part of the existing video index, thus, the hybrid approach tends to switch to the traditional ancestor traversal mode. Also, it can be seen as similar as the previous experiment that both the bulk-insertion algorithms outperform the one-by-one insertion algorithm.

5 Related Work

There exist a few techniques to improve the content-based video search. First, instead of processing videos in the traditional key-frames [7], there are many attempts have been proposed to summarize the content/feature of the videos for reducing the computational expense [3,8]. Subsequently, similarity measurement between two videos in the summarization form can be efficiently computed.

Once the features of the videos are extracted and summarized, indexing techniques are subsequently applied to further improve the efficiency of the video search [9]. For example, a B^+-tree is used to index the video summarization in [10]. In this work, each video is summarized in terms of the dominating content and changing-trend, called Bounded Coordinate System (BCS). Then, the first principle in PCA of the top-ranked features are selected and indexed. In [11], the authors proposed a hierarchical index structure called the Ordered VA-file (OVA-file), which is composed of a feature-vector file, a video-feature approximation file, and a slice summary file which points to the cluster in the approximation file. The OVA-file can improve the efficiency when the k-Nearest Neighbor (kNN) search is processed in video processing, i.e., to cluster the similar videos together. The proposed index structure can partition the whole index file into multiple slices such that only a small number of slices are accessed during the operation. Thus, the highly-efficient video content-based can be achieved. In our work, we focus on improving the efficiency of the bulk-insertion when the B^+-trees are applied to index the video databases, in which it can also be tailored to other types of the tree-structure index efficiently.

In the circumstances which the number of insertions is extremely high, the ability to insert more than one index entry at a time, is highly required. For the tree-structured indexes as the B^+-trees, not only the bulk should be inserted at a time, but also the structure of the tree after the insertion should be the same as insert the entries in one-by-one basis. Or, at least the tree should be still balance. For example, in [12], the authors proposed that the entries in the bulk should be used to create a sub-tree first. Subsequently, merge the sub-tree into the main index-tree. The merging-location is determined such that the bottom level of the sub-tree is at the bottom level of the pre-merged main tree. However, this approach needs the tree to be locked while the trees are being merged. The other notable approach to insert multiple index entries at a time is to re-built the whole tree which was proposed in [5]. The approach begins with sorting all the index entries by their keys increasingly. Subsequently, the pagination process, i.e., determine the index keys for the parents of the leaf nodes, is carried on. The

process will be continued until the root of the tree is determined recursively. However, it might not be applicable for our focus environment, but the off-line applications.

6 Conclusion

In this paper, we have proposed a hybrid bulk-insertion for the video content-based indexing. Based on the intuition that the range of the keys to be inserted in a bulk of videos might not be large, the leaf-level traversal is proposed to complement with the traditional ancestor traversal. The proposed approach works in a hybrid-manner, i.e., it switches from the ancestor traversal mode to the leaf traversal mode when the keys of the next leaf node can cover the remaining index entries. This can be achieved by a very small additional space complexity. The experiments have been conducted to evaluate our proposed work by comparing to the one-by-one insertion approach, and the traditional bulk-insertion approach. According to the experiments, we have found that the proposed hybrid bulk-insertion approach is more efficient than the traditional bulk-insertion approach, let alone the one-by-one insertion approach, which might not be appropriated for the video indexing in the current situation of video search applications. Though the efficiency of the proposed approach is different from the traditional approach drastically, given that it can be achieved by only $\Theta(n)$ additional space complexity, the approach can be useful in practices.

Our future work will focus on further investigating the video index bulk-insertion on distributed environments, e.g., Peer-to-Peer (P2P) network, in which the index update issues need to be addressed.

Acknowledgment. We would like to thank Faculty of Engineering, Chiang Mai University, Thailand, as well as the Graduate School for their financial support.

We also would like to thank Mr. Chaiyut Pradidtong-ngam for his help on system implementation, as well as other colleagues at Data Engineering Laboratory, Chiang Mai University, for their encouragement.

References

1. Wesch, M.: Youtube statistics (2008), http://mediatedcultures.net/ksudigg/?p=163
2. Cheng, R., Huang, Z., Shen, H.T., Zhou, X.: Interactive near-duplicate video retrieval and detection. In: MM 2009: Proceedings of the seventeen ACM International Conference on Multimedia, pp. 1001–1002. ACM, New York (2009)
3. Shen, H.T., Ooi, B.C., Zhou, X.: Towards effective indexing for very large video sequence database. In: SIGMOD 2005: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 730–741. ACM, New York (2005)
4. Zhou, X., Zhou, X., Shen, H.T.: Efficient similarity search by summarization in large video database. In: Bailey, J., Fekete, A. (eds.) Eighteenth Australasian Database Conference (ADC 2007), CRPIT, Ballarat, Australia, ACS, vol. 63, pp. 161–167 (2007)

5. Kim, S.W.: On batch-constructing b+-trees: algorithm and its performance evaluation. Information Sciences 144, 151–167 (2002)
6. Pollari-Malmi, K., Soisalon-Soininen, E.: Concurrency control and i/o-optimality in bulk insertion. In: String Processing and Information Retrieval, pp. 161–170 (2004)
7. Chang, H.S., Sull, S., Lee, S.U.: Efficient video indexing scheme for content-based retrieval. IEEE Transactions on Circuits and Systems for Video Technology 9, 1269–1279 (1999)
8. Cheung, S.S., Zakhor, A.: Efficient video similarity measurement with video signature. IEEE Transactions on Circuits and Systems for Video Technology 13, 59–74 (2003)
9. Böhm, C., Berchtold, S., Keim, D.A.: Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. ACM Computing Suveys 33, 322–373 (2001)
10. Huang, Z., Shen, H.T., Shao, J., Zhou, X., Cui, B.: Bounded coordinate system indexing for real-time video clip search. ACM Transactions on Information Systems 27, 1–33 (2009)
11. Lu, H., Ooi, B.C., Shen, H.T., Xue, X.: Hierarchical indexing structure for efficient similarity search in video retrieval. IEEE Transactions on Knowledge and Data Engineering 18, 1544–1559 (2006)
12. Chen, L., Choubey, R., Rundensteiner, E.A.: Merging r-trees: Efficient strategies for local bulk insertion. Geoinformatica 6, 7–34 (2002)

Author Index

Alor-Hernandez, Giner 27

Bae, Jae-Hak J. 126
Bain, Mike 247
Bai, Quan 14
Beydoun, Ghassan 111
Bindoff, Ivan 180

Cagalaban, Giovanni 103
Cai, Xiongcai 247
Chen, Tsung Teng 261
Choi, Sook-Young 91
Cho, Sung-Bae 223
Colomo-Palacios, Ricardo 27
Compton, Paul 135, 150, 247
Cortes-Robles, Guillermo 27

Dazeley, Richard 195, 235

Fareed, Umer 300
Finlayson, Angela 135
Fushimi, Takayasu 208

Gomez-Berbis, Juan Miguel 27
Guldrís-Iglesias, Fernando 27

Ha, Sung Ho 82
Hoffmann, Achim 165
Huang, Xu 288

Itoh, Hidenori 49
Ito, Takayuki 14

Jimenez-Domingo, Enrique 27
Johnson, Scott 195
Jung, Tae-min 223

Kang, Byeong H. 235
Kang, Byeong Ho 180
Kato, Shohei 49
Kawamura, Takahiro 61
Kelarev, Andrei V. 235
Kim, Mihye 91
Kim, Sang-Rak 126
Kim, Seoksoo 103
Kimura, Masahiro 208, 273
Kim, Yang Sok 247
Krzywicki, Alfred 247
Kwon, Eun Kyoung 82

Lee, Maria R. 261
Lee, Young-Seol 223
Lim, Andrew 38

Mahidadia, Ashesh 247
Masuda, Megumi 49
Misra, Avishkar 150
Motoda, Hiroshi 208, 273

Nakagawa, Hiroyuki 61
Natwichai, Juggapong 310
Nguyen, Phuoc 288

Ohara, Kouzou 208, 273
Ohsuga, Akihiko 61
Okatani, Kazuhiro 1
Onkhum, Narissa 310
Othman, Siti Hajar 111

Posada-Gomez, Rubén 27

Qin, Hu 38

Rodríguez-González, Alejandro 27

Saito, Kazumi 208, 273
Sharma, Dharmendra 288
Sowmya, Arcot 150

Tahara, Yasuyuki 61
The, Nguyen Minh 61
Tran, Dat 288

Vamplew, Peter 195

Warner, Philip 195
Wobcke, Wayne 247

Xu, Han 165
Xu, Shuxiang 73
Xu, Zhou 38

Yang, Jae-Gun 126
Yearwood, John L. 235
Ye, Dayong 14
Yoshida, Tetsuya 1
Yoshikawa, Yuya 273

Zhang, Byoung-Tak 300
Zhang, Minjie 14
Zhu, Wenbin 38